Net Zero Carbon Campaign Smart Contract
For Information Communication Technology Sectors to Achieve Net Zero Targets Through Science Based Target Initiatives Approached

Siddhesh Deshmukh, Sonny Dhamayana, Maksim Garbuzov, Jelena Gjorgjev
Global Master in Blockchain Technologies
Zigurat Global Institute of Technology Barcelona, Spain

Abstract

Net Zero Carbon Campaign Smart Contract is one of proactive initiatives to guide and lead Information Communication Technology Business Sector to achieve Net Zero Emission Target through Science Based Target (SBTi) Methodology. The campaign smart contract integrated to Decentralized Application developed in 3rd Generation Blockchain Ecosystem using Plutus Language which further deployed in Cardano Blockchain Main Network.

This application program developed as part offset ambitious emissions reductions targets in line with the latest climate science. It is focused on leading and accelerating ICT companies across the world to halve emissions before 2030 and achieve net-zero emissions before 2050 through global reward Net Zero Campaign to achieve scenarios that limit warming to 1.5°C with no or limited overshoot reach net zero CO2 emissions around 2050, accompanied by rapid reductions in non-CO2 GHG emissions.

Keywords: Net Zero Carbon, SBTi, Smart Contract, Plutus, Campaign

Part 1: **Coding a Blockchain**

1. Net Zero Carbon Campaign Smart Contract - Introduction

The IPCC (2018), Special Report on Global Warming of 1.5°C (SR15), was widely accepted as a warning that we must limit global temperature rise to 1.5°C above pre-industrial levels and reach net-zero CO2 emissions by 2050 for the best chance of avoiding catastrophic climate breakdown. More recently, the IPCC (2021), Sixth Assessment Report, has confirmed that climate change is already affecting every region on Earth, its impacts increasingly visible in the form of extreme weather, worsened droughts, and heightened risk of forest fires. Against this backdrop, companies are increasingly adopting net-zero targets.

The number of businesses committing to reach net-zero emissions has grown rapidly, but not all net-zero targets are equal. Without adhering to a common definition, net-zero targets can be inconsistent, and their collective impact is strongly limited. While the growing interest in net-zero targets represents an unparalleled opportunity to drive corporate climate action, it has also created a pressing need for a common understanding of “net-zero” in a corporate context.

Business leaders need a robust, science-based framework for setting net-zero targets. Otherwise, they risk continuing to invest in business models that are inconsistent with the goals of the Paris Agreement. Through a transparent multi-stakeholder process, the Science Based Targets (Science Based Targets, 2021) initiative (SBTi) has developed the first global science-based standard for companies to set net-zero targets.

The Net-Zero Standard gives business leaders confidence that their near-term and long-term targets are aligned with what is needed to contribute to a habitable planet, and it provides clarity on business climate action to a wide range of stakeholders. Through the SBTi, companies can commit to net-zero, which includes setting validated near-term and long-term science-based targets consistent with limiting temperature rise to 1.5°C, to become distinguished as climate leaders and drive forward the global transition to net-zero.

The Science Based Target Initiatives (SBTi) is a global body enabling businesses to set ambitious emissions reductions targets in line with the latest climate science. It is focused on accelerating companies across the world to halve emissions before 2030 and achieve net-zero emissions before 2050.

The initiative is a collaboration between CDP, the United Nations Global Compact, World Resources Institute (WRI) and the Worldwide Fund for Nature (WWF) and one of the We Mean Business Coalition commitments.

The SBTi defines and promotes best practice in science-based target setting, offers resources and guidance to reduce barriers to adoption, and independently assesses and approves companies' targets.

The SBTi's Corporate Net-Zero Standard (also referred to as the Net-Zero Standard) provides guidance, criteria, and recommendations to support corporates in setting net-zero targets through the SBTi (Anon., 2021).

The main objective of this standard is to provide a standardized and robust approach for corporates (Science Based Targets, 2021) to set NetZero targets (Science Based Targets, 2021) that are aligned with climate science. It is important to note that while the SBTi does provide some supplementary guidance on greenhouse gas (GHG) accounting, companies should refer to the suite of corporate Greenhouse Gas Protocol standards on this topic.

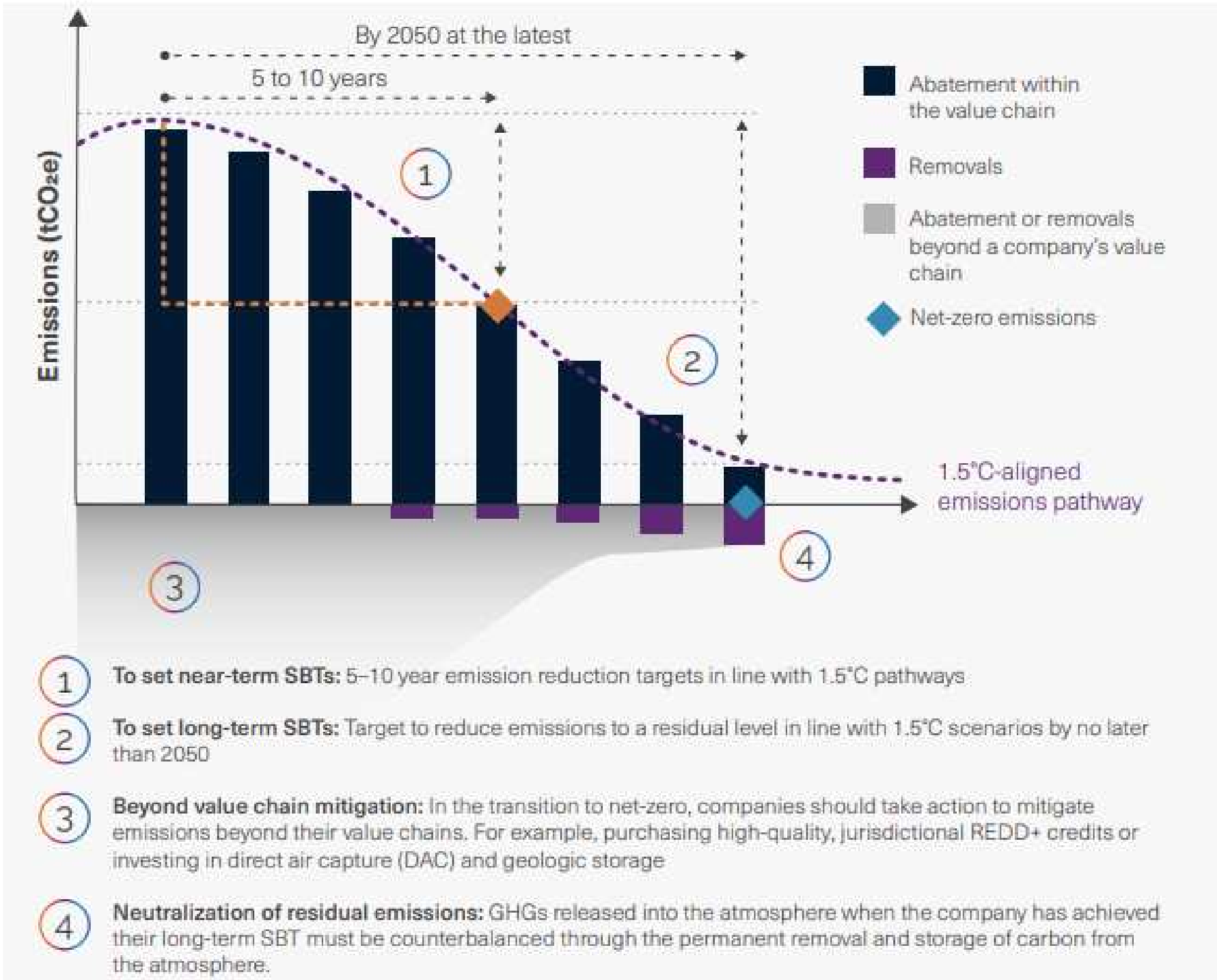

Figure 1 Key elements of the Net-Zero Standard

Previously Near Term SBT's known as "science-based targets", these are 5-10-year GHG mitigation targets in line with 1.5°C pathways. When companies reach their near-term target date, they must calculate new near-term science-based targets to serve as milestones on the path towards reaching their long-term science-based target. Near-term science-based targets galvanize the action required for significant emissions reductions to be achieved by 2030. Near-term emissions reductions are critical to not exceeding the global emissions budget and are not interchangeable with long-term targets.

Long Term SBT's show companies how much they must reduce value chain emissions to align with reaching net-zero at the global or sector level in eligible 1.5°C pathways by 2050 or sooner. These targets drive economy-wide alignment and long-term business planning to reach the level of global emissions reductions needed for climate goals to be met based on science. A company cannot claim to have reached net-zero until the long-term science-based target is achieved.

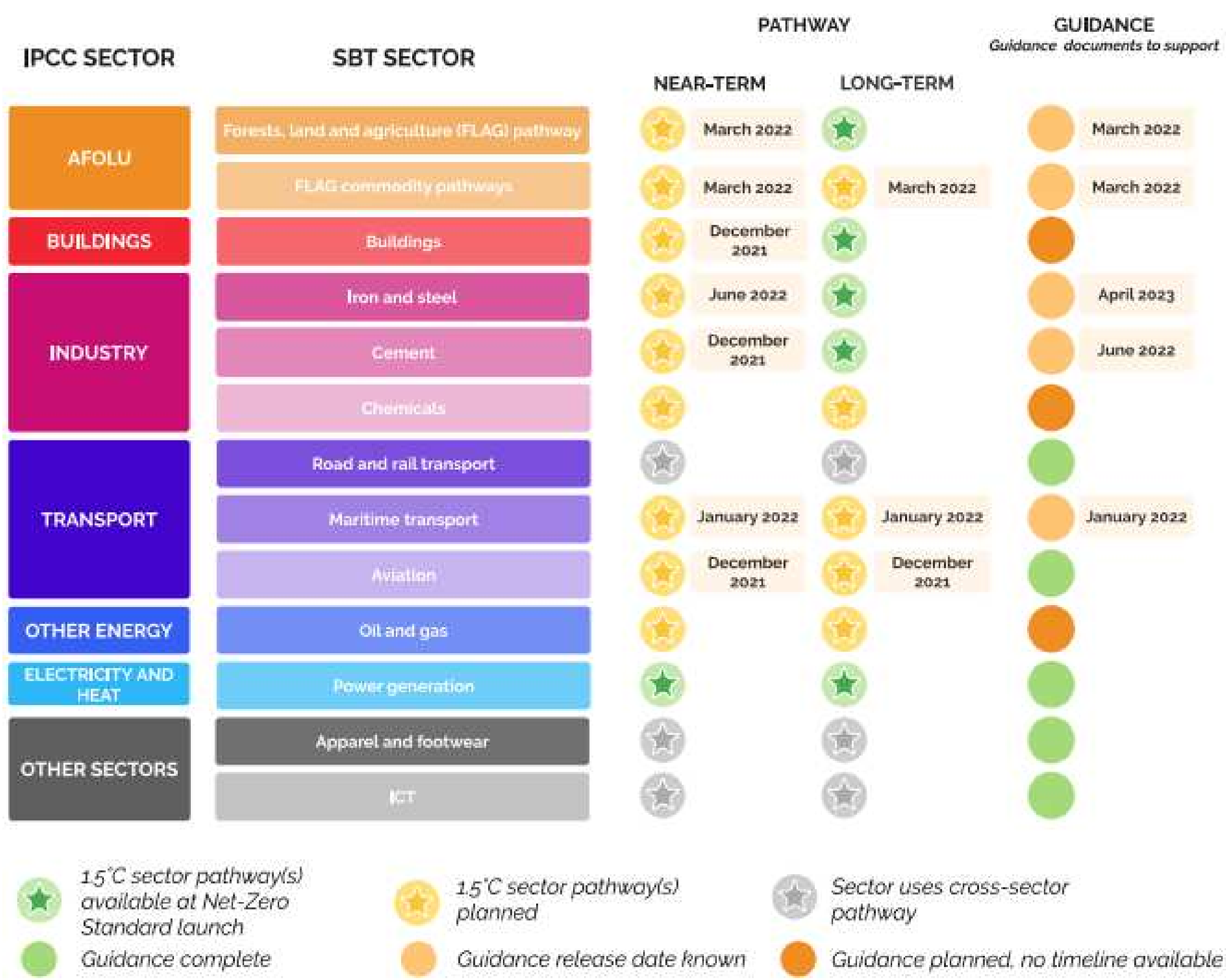

Figure 2 Summary of the status of sector-specific guidance and pathways

Companies can take a variety of approaches to developing near-term and long-term science-based targets; however, the SBTi recommends following the five steps.

Figure 3 The SBTi recommends a five-step approach to setting science-based targets.

Companies need to establish a base year to track emissions performance consistently and meaningfully over the target period. The following considerations are important for selecting a base year:

- Scope 1, 2, and 3 emissions data should be accurate and verifiable.
- Base year emissions should be representative of a company's typical GHG profile3.
- The base year should be chosen such that targets have sufficient forward-looking.
- ambition.
- The base year must be no earlier than 2015.

Companies that have already set near-term science-based targets must use the same base year for their long-term science-based target.

LONG-TERM TEMPERATURE GOAL	ANNUAL LINEAR REDUCTION RATE OVER TARGET PERIOD
Well-below 2˚C Approx. 66% chance of limiting peak warming between present and 2100 to below 2˚C.	$2.5\% \leq X < 4.2\%$
1.5˚C Approx. 50% chance of limiting warming in 2100 to 1.5˚C.	$X \geq 4.2\%$

Table 1 Ambition ranges for target classification of near-term science-based targets

The following steps should be adopted to set a science-based target by a company with ICT operations (Science Based Targets, 2021):

- Select a baseline year.

  The SBTi recommends using the most recent year for which data is available.

Note: 2015 is the baseline year for the sector and sub-sectors trajectories. Thus, companies may also be interested in monitoring their yearly emissions compared to 2015 levels if these 2015 emissions are readily available.

- Select a target year.

  Targets must cover a minimum of 5 years. Due to the fast-changing nature of digital technologies, this guidance recommends that ICT companies should set a target year no further ahead than 2030. In any case, it is worth noting that the SBTi criteria require companies to review, and, if necessary, revalidate their targets every five years from the date of the original target approval

- Measure Scope 1 and 2 Emission

  Scopes 1 and 2 emissions need to be measured for the baseline year. They need to be measured according to the GHG Protocol using a common boundary approach across all company operations. Most companies will have activities in addition to their ICT operations, such as office buildings and / or a transport fleet. In such cases, companies may choose to combine all their scope 1 and 2 emissions and derive a single SBT following the ICT sector method, thereby allowing the overall trajectory to stay within an ambitious 1.5°C trajectory. This is the simplest approach and at this point it is the recommended alternative as it keeps the company consistent with a 1.5°C trajectory for its overall operation. However, companies may also wish to establish a separate SBT associated with the scope 1 and 2 emissions arising from their support activities using, for example, the SBTi's existing relevant SDA methodology and tools.4 In this case, only the Scope 1 and 2 emissions associated with operating the ICT equipment should be used in the next step.

- Calculating the science-based target.

  A sub-sector science-based target (SBT's) is then calculated by multiplying the combined Scope 1 and 2 emissions in the base line year (CCb) by an emissions reduction factor (ERF). The emissions reduction factor is based on the appropriate sub-sector emission reduction pathway (see section 2.2 Figure 2) and the baseline and target years. Emission Reduction Factor values for mobile, fixed and data center sub-sectors, and for different baseline and target years compatible with section 2.2 are listed in Annex C. Annex A explains how these factors were derived and provides further guidance on Scope 1-2 emissions.

- Worked examples on calculating an ICT sub-sector target.

$$SBT_s = CC_b \cdot ERF$$

Equation 1 Calculating an ICT sub-sector target

A sub-sector science-based target (SBTs) is then calculated by multiplying the combined emissions in the base line year (CCb) by an emissions reduction factor (ERF). The emissions reduction factor is based on the appropriate sub-sector emission reduction pathway and the baseline and target years.

Emission Reduction Factor values for mobile, fixed and data center sub-sectors, and for different baseline and target years describe as below table.

C1 Mobile Networks

		Target Year							
		2023	2024	2025	2026	2027	2028	2029	2030
Base Year	2018	0.863	0.824	0.786	0.736	0.686	0.636	0.587	0.537
Base Year	2019		0.833	0.794	0.744	0.694	0.643	0.593	0.543
Base Year	2020			0.803	0.752	0.701	0.650	0.599	0.548
Base Year	2021				0.783	0.730	0.677	0.624	0.571
Base Year	2022					0.761	0.706	0.651	0.595
Base Year	2023						0.737	0.680	0.622
Base Year	2024							0.712	0.651
Base Year	2025								0.683

Table 2 Emission Reduction Factor C1 Mobile Networks

C2 Fixed Networks

		Target Year							
		2023	2024	2025	2026	2027	2028	2029	2030
Base Year	2018	0.764	0.701	0.638	0.582	0.526	0.470	0.415	0.359
Base Year	2019		0.717	0.652	0.595	0.538	0.481	0.424	0.367
Base Year	2020			0.668	0.609	0.551	0.493	0.434	0.376
Base Year	2021				0.653	0.590	0.528	0.465	0.403
Base Year	2022					0.636	0.568	0.501	0.434
Base Year	2023						0.615	0.543	0.470
Base Year	2024							0.592	0.512
Base Year	2025								0.563

Table 3 Emission Reduction Factor C2 Fixed Networks

C3 Data Centres

		Target Year							
		2023	2024	2025	2026	2027	2028	2029	2030
Base Year	2018	0.809	0.755	0.700	0.651	0.603	0.554	0.505	0.456
Base Year	2019		0.765	0.710	0.660	0.611	0.561	0.512	0.463
Base Year	2020			0.720	0.669	0.619	0.569	0.519	0.469
Base Year	2021				0.709	0.656	0.603	0.550	0.497
Base Year	2022					0.698	0.641	0.585	0.528
Base Year	2023						0.684	0.624	0.564
Base Year	2024							0.669	0.605
Base Year	2025								0.652

Table 4 Emission Reduction Factor C3 Data Centers

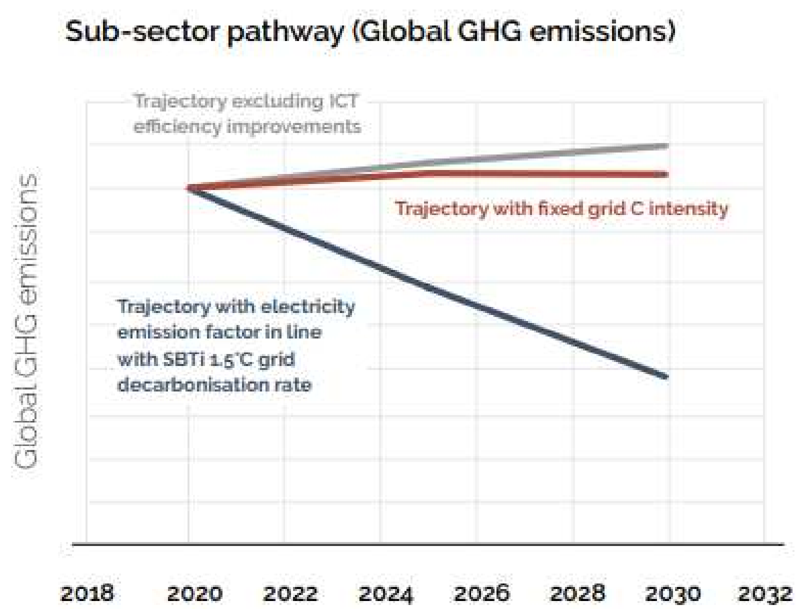

Figure 4 Sub Sector Global GHG Emission Path Way

For ICT operators, emissions associated with the generation and subsequent use of electricity dominate their combined footprints where the pathway associated with these emissions would therefore be expected to be like that of the power sector which actually reaches a zero-convergence point by 2050.

A mathematical analysis of the implications of such a zero-convergence points which results in the following absolute, rather than intensity-based, target trajectory according to:

$$C_{SBT,y} = CC_b \cdot SC_{s,y} \ / \ SC_{s,b}$$

Equation 2 Implications of such a zero-convergence points

1. CCb CO2 emissions of company in base year b (tCO2)
2. SCs,b CO2 emissions of sub-sector "s" in base year b (tCO2)
3. SCs,y CO2 emissions of sub-sector "s" in year y (tCO2)

Net Zero Carbon Campaign Smart Contract initiatives which focused on ICT sectoral target-setting approach does not follow the usual intensity approach but follows a simplified absolute approach.

The first step in establishing a sectoral target-setting approach is to identify an appropriate activity metric. For example, in the case of the power sector, activity levels are measured by the number of MWhrs of electricity generation each year. For commercial buildings it is the floor area in m2 of real estate for a given year. Initially it was assumed that the ICT sector would also adopt an intensity model. As it was considered very unlikely that there could be a single form of activity metric relevant to all ICT sub-sectors, a sub-sector approach was adopted. Based on the mathematical equations presented in the SBTi SDA methodology report, an ICT sub-sector pathway associated with use-phase electricity. The outcome is to get validate data information of Carbon emissions in year y (tCO2).

The 1st Genesis Model of Net Zero Carbon Campaign Smart Contract defined as simple bounty algorithm with Fund Rewards which will be given to the winner of the campaign or to the registered participant who can achieve Carbon Emission below the Science Based Target on the Target Year.

The Simplified Net Zero Carbon Campaign Smart Contract which developed through Plutus Language consists of basic components are the properties, the logic and the ledger. Each of these components can be mapped directly into technical concepts. Properties represent a data schema, logic represents code, and the ledger corresponds to a database.

Further development, Net Zero Carbon Campaign Smart Contract Version 2 have been upgraded by added function the requirement for campaign participant to deposit minimum fund contribution, Token reward has been added inclusive with fund rewards, added of some error messages, adding validator nodes, adding of campaign start function, and adding of campaign ended function.

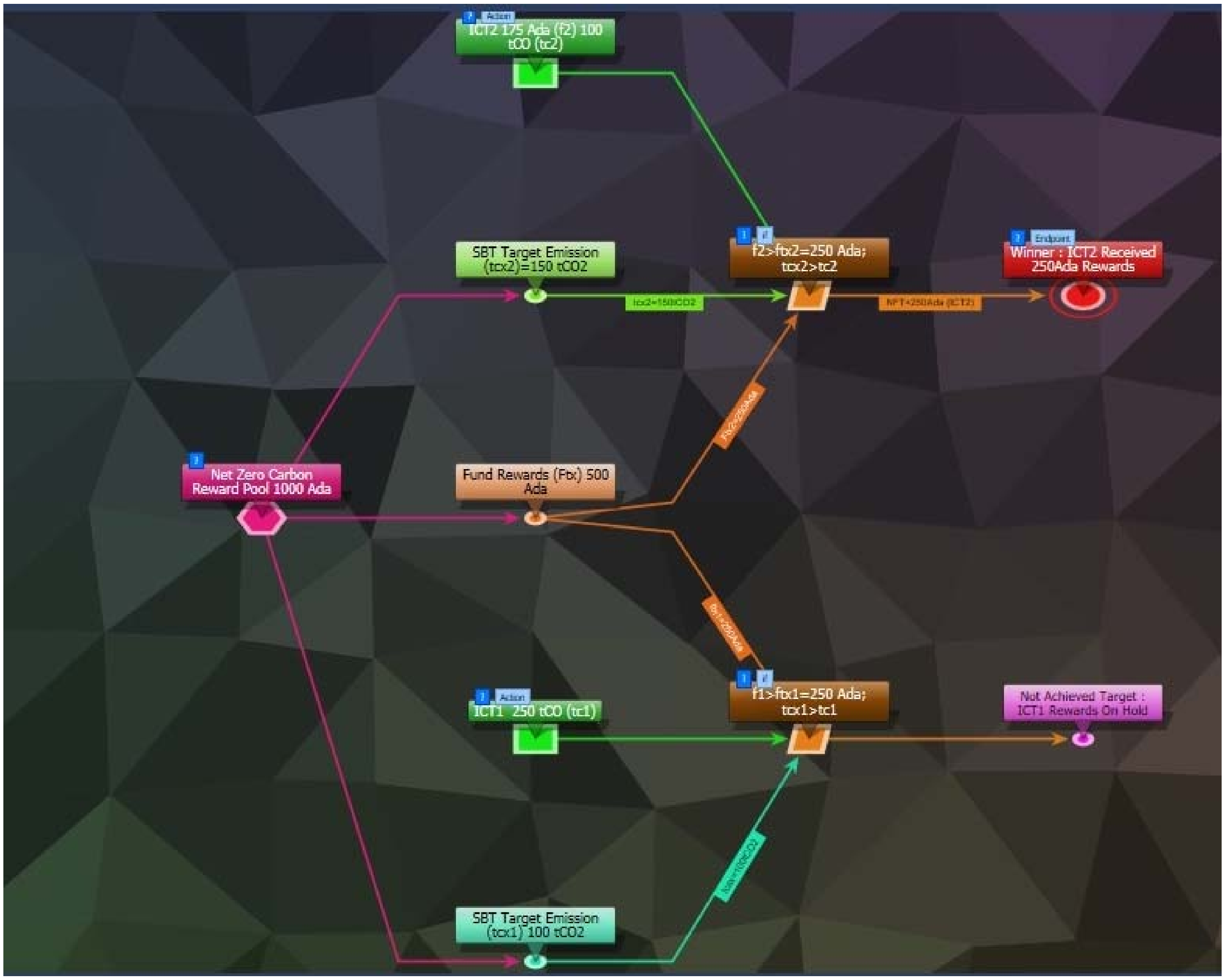

Figure 5 SC Genesis - Version 1 (Bounty)

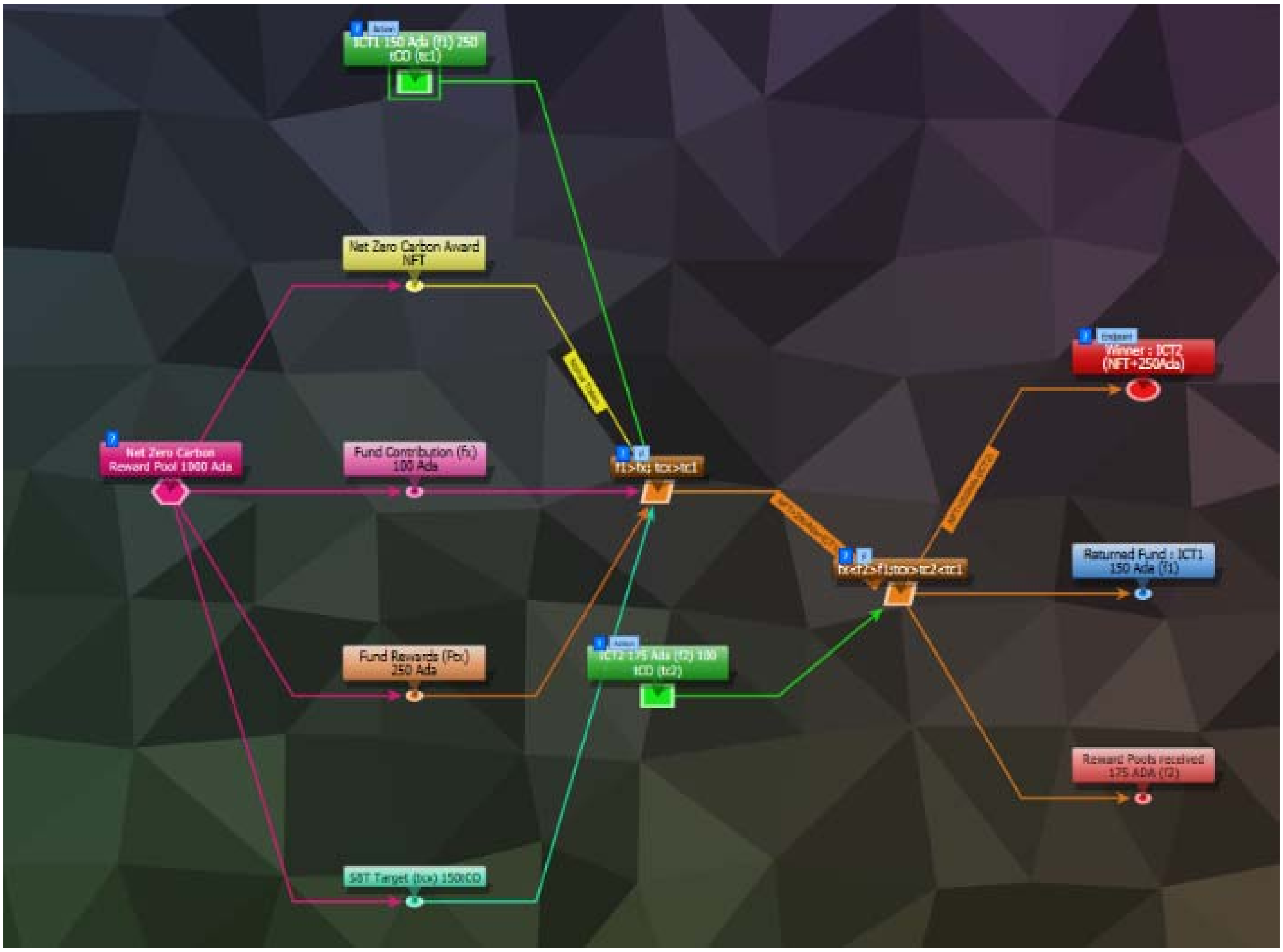

Figure 6 SC Version 2 (Bounty Auction)

2. Net Zero Carbon Campaign Smart Contract – Algorithm and Features

In the 1st development of Net Zero Carbon Campaign Smart Contract which called as Version 1.0 or Genesis Version, the Smart Contract developed in simple Campaign which adopt Bounty Games algorithm. The Campaign Pools will set Participant Public Key Beneficiary Address, the Campaign Baseline Year, Campaign Target Year, tCO2 target emission based on Science Based Target Calculation, target Emission Reduction Factor, Fund Rewards, and pin authorization. All the settings are configured as 1 (one) participant as 1 (one) dedicated Campaign. This is to ensure safe and secure Rewards Funds will only be distributed to each participant who achieves target emission on target year.

Further development in Version 2.0, there are some adjustments and upgrades in the algorithm which adopt combination of Bounty Games and Auction (IOHK, 2023). These algorithms upgrade as part stage development to bring Campaign Pools become Carbon Trading Pools. The new arrangements such as Campaign Start, Campaign End, and Participant Submission Report will be part of this upgrade.

Campaign Start Smart Contract Schema is used to define and set up the campaign duration, minimum participant fund contribution, maximum CO2 emission target, currency, and reward token.

Campaign Ended Smart Contract Schema is used to close the campaign when the elapsed time of campaign duration has ended.

Participant Submission Report Smart Contract Schema is used for campaign participants to register their participation through paid the fund contribution and submit their emission report. The fund contribution will be valued as correspondence to participant report submission, which will be higher price if the reports have been validated and auditable. In future development the validation will go through validation node which involves generative AI emission validator and trustee registered auditor.

To deep dive the feature algorithm and how these smart contract works will be defined as below:

2.1 Net Zero Carbon Campaign Smart Contract Version 1.0 (Genesis)

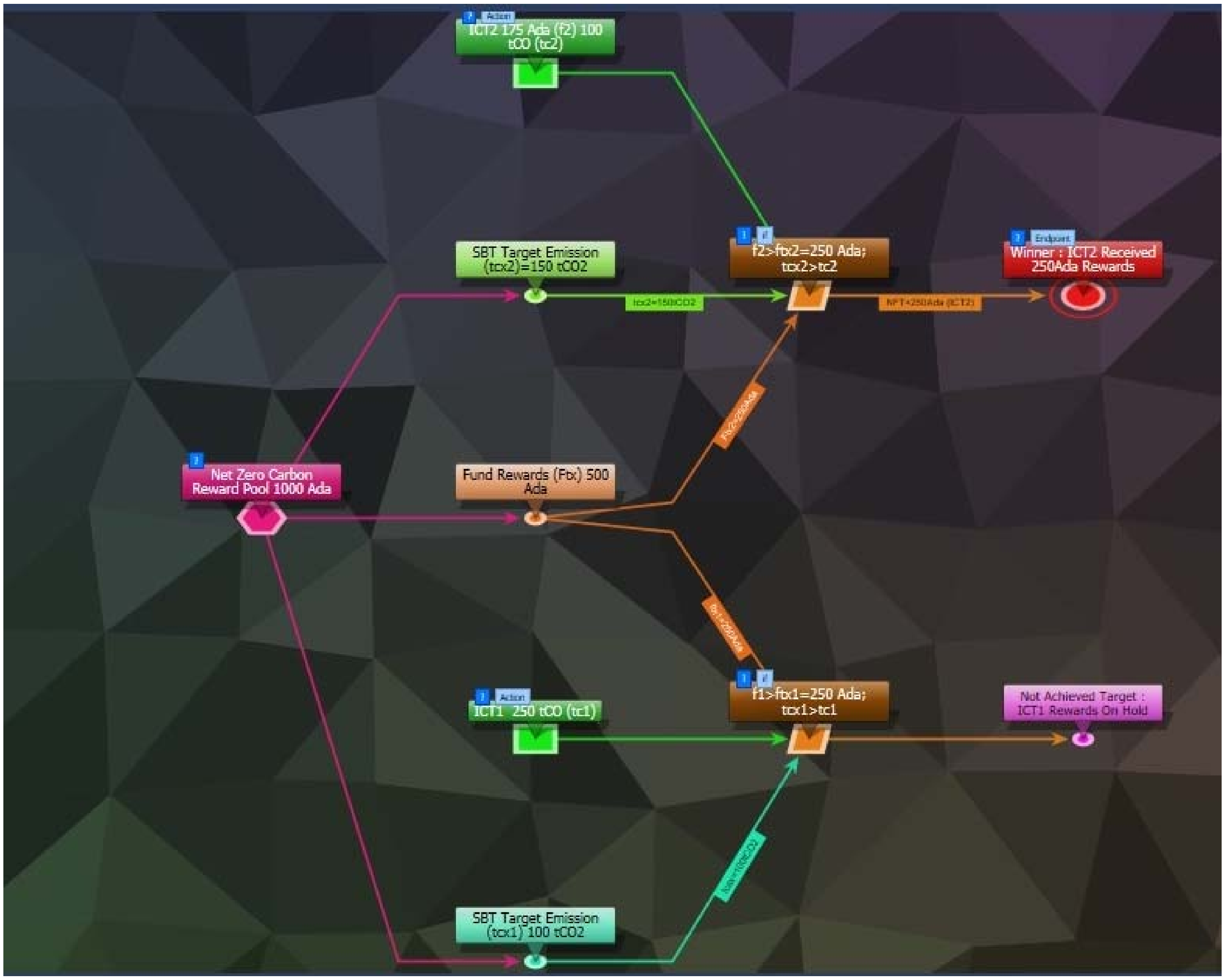

Figure 7 Net Zero Carbon Campaign Smart Contract Genesis -Version 1 (Bounty)

The simple features logic algorithm of Net Zero Carbon Emission Campaign based on Science Based Target as part genesis algorithm development defined as:

2.1.1 On-chain Smart Contract, which consists of:

- Carbon Credit SBT Input Data Script Datum and Redeemer (On Chain)

Datum is the locker's input Data Script which will send a datum hash by locker algorithm when locking. Meanwhile Redeemer is the unlocker’s input Data Script. The Unlocker must send the datum matching the hash and any redeemer Data Script Input when unlocking algorithm is processed. The Unlocker must be made aware of the datum where the locker used to make the hash.

```
-- Input Data Script Locker (Datum)
data SBTDatum = SBTDatum
-- PublicHashKey Beneficiary Address
{ beneficiary      :: !PaymentPubKeyHash
-- Baseline Year Science Based Target
, set_BaselineYear :: !Integer
-- Target Year Science Based Target
, set_TargetYear   :: !Integer
-- SBT CO2 Emission in Tonnage
, set_SBTTonCO2E   :: !Integer
-- Target Year ERF
, set_ERF          :: !Integer

-- Carbon Emission Reward Amount
, set_rewardAmount :: !Integer
-- Pin Reward Authorization
, set_pin          :: !Integer
} deriving Show
PlutusTx.unstableMakeIsData ''SBTDatum
PlutusTx.makeLift ''SBTDatum

-- Input Data Script Unlocker (Redeemer)
data CCRewardRedeemer = CCRewardRedeemer
-- Validated Value of Current Year Carbon Emission
{ val_cc      :: !Integer
-- Validated Value 0f Current Year ERF
, val_erf     :: !Integer
-- Validated Pin Reward Withdrawl Authorization
, val_pin     :: !Integer
} deriving Show
PlutusTx.unstableMakeIsData ''CCRewardRedeemer
PlutusTx.makeLift ''CCRewardRedeemer
```

Source Code 1 Carbon Credit SBT Input Data Script Datum and Redeemer (On Chain)

- Carbon Credit SBT Validation Function On Chain

Generally, Plutus validators are functions that have three inputs which consist of the Datum, the Redeemer, and the Script Context where further return a Boolean.

$$validator :: Datum \rightarrow Redeemer \rightarrow ScriptCxt \rightarrow Bool$$

Plutus Smart contracts are executed implicitly when UTxOs at script addresses are used as inputs in a transaction. The script context provides information about the pending transaction, along with which input triggered the validation.

The datum and redeemer are user input, but crucially they are different users input which is provided by the same user. There are two types of users for a Plutus segue. The user that creates the first transaction is the "locker". The user the creates the second transaction is the "unlocker". The datum is the locker's input. The redeemer is the unlocker's input.

validator :: LockerInput -> UnlockerInput -> ScriptCxt -> Bool

How these inputs are used, consider a swap situation where the locker, locks assets at a script address and provides a datum that describes how they would like to be paid. Then the unlocker uses the UTxO as input and passes in a redeemer explaining how they would like to be paid.

The validator looks at the two user's inputs and the script context to make sure the inputs and outputs are consistent with what is requested by the users. If they are, it returns to "true" and performs the swap. Otherwise, the transaction fails.

This conceptual understanding of datum as the locker's input and the redeemer as unlocker's input, is very useful when designing validators. However, it is not how Plutus transactions are constructed. The locker sends a datum hash when locking assets at a script address. The unlocker must send the redeemer and the full datum, such that it matches the hash the locker used when locking assets.

The unlocker must send both inputs, the locker's input and their own. Because the locker set the hash during locking, the unlocker is not able to send any datum. It must be the datum the locker chose. However, there is no on-chain mechanism to propagate the lockers choice to the validator. The unlocker must discover through a custom off-chain process what datum the locker used, otherwise unlocking is impossible.

As part Genesis Smart Contract, the validation algorithm utilized 4 (four) input Datum which are Public Key Address (phk), Setting Carbon Emission Target (ssbt), Setting Emission Reduction Factor (serf), and Pin Authorization (dpnt). Meanwhile for input Redeemer used Participant

Validation Carbon Emission Report (valcc), Validated Emission Reduction Factor in the current year (valerf), and Validated Pin Authorization (vpnt).

Further as part of Script Context, the validation will go through verified target emission achievement, pin authorization, and Signed Beneficiary Address. If all conditions have met, the Rewards fund will be released to the participant. In the condition, participant carbon emission report submission is above the target which has been set, the hash error "Emission Target Not Achieved - Deposits Fund on Hold" will be generated. Similarly, in the condition the participant put the wrong pin authorization, the hash error "Wrong Withdrawal Pin!" will be generated in On-chain Networks. The participant which are not registered their public key by Campaign Pools, even their submission has been achieved the target and put correct pin authorization, the submission will not be accepted and will receive hash error "Beneficiary's Signature Not Matched". This smart contract script to ensure that campaign run in safe and secure process where's that only the registered participant and the correct achiever will receive the rewards fund.

```
-- Carbon Credit Emission Validation On-Chain Code
{-# INLINABLE validateCCEmission #-}
validateCCEmission :: SBTDatum -> CCRewardRedeemer -> ScriptContext -> Bool
validateCCEmission (SBTDatum phk _ _ ssbt serf _ dpnt) (CCRewardRedeemer
valcc   valerf vpnt) ctx =
  traceIfFalse "Emission Target Not Achieved -  Fund Rewards On Hold" $
                valcc*valerf >= ssbt * serf  &&
  (traceIfFalse "Wrong Withdrawl Pin!" $ dpnt == vpnt) &&
  traceIfFalse "Beneficiary's Signature Not Matched" signedByBeneficiary
where
      info :: TxInfo
      info = scriptContextTxInfo ctx
      signedByBeneficiary :: Bool
      signedByBeneficiary = txSignedBy info $ unPaymentPubKeyHash $ phk
```

Source Code 2 Carbon Credit SBT Validation Function On Chain

- Data Script Wrapped Compiler Function

Datum, Redeemer, and Context Script Wrapped Compiler is where the compilation to Plutus Core happens. This compiler utilized something called Template Haskell to take the above Haskell function and compile it to Plutus Core.

```
-- Datum and Redeemer parameter types
data CCRewards
instance Scripts.ValidatorTypes CCRewards where
      type instance DatumType CCRewards = SBTDatum
      type instance RedeemerType CCRewards = CCRewardRedeemer

-- The script instance to compile validator (ready to go onto the chain)
ccRewardsInstance :: Scripts.TypedValidator CCRewards
ccRewardsInstance = Scripts.mkTypedValidator @CCRewards
      $$(PlutusTx.compile [|| validateCCEmission ||])
      $$(PlutusTx.compile [|| wrap ||])
where
      wrap = Scripts.wrapValidator @SBTDatum @CCRewardRedeemer
```

Source Code 3 Data Script Wrapped Compiler Function

2.1.2 Off-chain Smart Contract, which consists of:

- Carbon Credit SBT Parameter Input Data and Validation Report Parameter (Off-Chain)

Off-Chain Input Data defined as input parameter which segregated as Science Based Target Set Parameter (SBTParams) and Carbon Emission Submission Report Parameter (CCEmissionParams).

SBTParams Data consists of Participant Public Hash Key Beneficiary Address (rewardBeneficiary), Campaign Baseline Year (baselineYear), Campaign Target Year (targetYear), Science Based Target Carbon Emission in tCO2 (target_SBTTonCO2E), Target Emission Reduction Factor (target_ERF), Rewards Amount (amount_Rewards), and Pin Authorization (pin).

```
-- | The Address of the Carbon Credit Science Based Target Rewards
ccRewardsAddress :: Address
ccRewardsAddress=Ledger.scriptAddress (Scripts.validatorScript
                  ccRewardsInstance)
-- | Parameters for the "Carbon Credit SBT Rewards" endpoint
data SBTParams = SBTParams
-- Public Hash Key Beneficiary Address
{ rewardBeneficiary  :: !PaymentPubKeyHash
-- Baseline Year Science Based Target
, baselineYear       :: !Integer
-- Target Year Science Based Target
, targetYear         :: !Integer
-- Science Based Target CO2 Emission in Tonnage
, target_SBTTonCO2E  :: !Integer
-- Target Year Emission Reduction Factor
, target_ERF         :: !Integer
-- Carbon Emission Reward Amount
, amount_Rewards     :: !Integer
-- Pin Reward Withdrawl Authorization
, pin                :: !Integer
}
deriving (Generic, ToJSON, FromJSON, ToSchema)
```

Source Code 4 Carbon Credit SBT Parameter Input Data and Validation Report Parameter (Off-Chain)

CCEmissionParams Data consist of Validated Current Emission Submission Report in ton CO2 (val_Current_Emission_TonCO2E), Validated Current Emission Reduction Factor (val_Current_Emission_Reduction_Factor), and Pin Validation (pin_validation).

```
--  Parameters for the "Carbon Emission Validation" endpoint
data CCEmissionParams = CCEmissionParams
-- Validated Value of Current Year Carbon Emission in Tonnage
{ val_Current_Emission_TonCO2E           :: !Integer
-- Validated Value 0f Current Year Emission Reduction Factor
, val_Current_Emission_Reduction_Factor  :: !Integer
-- Validated Pin Reward Withdrawl Authorization
, pin_validation                         :: !Integer
}
deriving stock (Prelude.Eq, Prelude.Show, Generic)
deriving anyclass (FromJSON, ToJSON, ToSchema, ToArgument)
```

Source Code 5 Carbon Credit SBT Parameter Input Data and Validation Report Parameter (Off-Chain)

- Schema End Points

This Schema is to tie up all the code in Off Chain Smart Contract End Points with one endpoint.

```
-- The schema of the smart contract which consolidated all end points
type CCRewardsSchema =
     Endpoint "Science Base Target GHG Data Center and Funds Deposit"
      SBTParams
 .\/ Endpoint "Updated Current GHG Data Center and Released Funds Deposit"
      CCEmissionParams
```

Source Code 6 Schema End Points

- Science Based Target Carbon Emission Contract End Point Off-Chain

SBT Carbon Emission Smart Contract End Point Off-Chain is defined as Smart Contract which runs on Off-Chain. The program consists of Contract Error Log Info, Data Script Input Parameter, Contract Scripts, and Reward Schema. This contract end point defined all the setup parameters as locked datum prior to releasing the rewards.

```
-- The "Carbon Credit Science Based Target Rewards" contract endpoint.
ccsbtrewards :: AsContractError e => SBTParams > Contract ()
                   CCRewardsSchema e ()
ccsbtrewards (SBTParams bnf baseyear targetyear sbttarget erftarget
                   rewardAmt pnt ) = do
let datDatum = SBTDatum
      -- SBT Datum Public Hash Key Beneficiary Address
            { beneficiary       = bnf
       -- SBT Datum Baseline Year Science Based Target
          , set_BaselineYear  = baseyear
      -- SBT Datum Target Year Science Based Target
          , set_TargetYear    = targetyear
      -- SBT Datum Science Based Target CO2 Emission in Tonnage
          , set_SBTTonCO2E    = sbttarget
      -- SBT Datum Target Year Emission Reduction Factor
          , set_ERF           = erftarget
      -- SBT Datum Carbon Emission Reward Amount in ADA
          , set_rewardAmount  = rewardAmt
      -- SBT Datum Pin Reward Withdrawl Authorization
          , set_pin           = pnt
          }
```

```
let tx    = Constraints.mustPayToTheScript datDatum $ Ada.lovelaceValueOf
            rewardAmt
ledgerTx <- submitTxConstraints ccRewardsInstance tx
void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
logInfo @String $ printf "Science Based Target Carbon Credit Emission
      Rewards with Baseline Year %d and Target Year %d"  baseyear
      targetyear
logInfo @String $ printf "Science Based Target Carbon Credit Emission
      is %d TonCO2e"  (sbttarget * erftarget)
logInfo @String $ printf "Carbon Credit Bonus Available Funds of %d
      and Token Rewards to be credited to Company with below Carbon
      Emission SBT Achievement on Target Year" rewardAmt
```

Source Code 7 Science Based Target Carbon Emission Contract End Point Off-Chain

- Science Based Target Carbon Emission Contract Validation End Point Off-Chain

This Smart Contract End Point Off-Chain defined all the setup parameters to unlock and released the rewards to the participant who achieved the target which has been set.

The smart contract consists of Contract Error Log Info, Data Script Input Parameter, Contract Scripts, and Reward Schema.

```
-- The "Carbon Emission Validation" contract endpoint.
carbonemissionupdate :: AsContractError e => CCEmissionParams
                         -> Contract () CCRewardsSchema e ()
carbonemissionupdate (CCEmissionParams valccemission targeterfvalue
      pin_validation) = do
onow   <- currentTime
opkh   <- ownPaymentPubKeyHash
-- filter all incorrect datum ccsbtrewards scripts
unspentOutputs <- Map.filter hasCorrectDatum <$> utxosAt ccRewardsAddress
let datRedeemer = CCRewardRedeemer
      -- Carbon Emission Validated Value of Current Year in Tonnage
             { val_cc    = valccemission
     -- Emission Reduction Factor Validated Value 0f Current Year
          , val_erf   = targeterfvalue

      -- Pin Reward Withdrawl Authorization Validated
          , val_pin   = pin_validation
          }
```

```
let tx = collectFromScript unspentOutputs datRedeemer
ledgerTx <- submitTxConstraintsSpending ccRewardsInstance unspentOutputs tx
void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
logInfo @String $ printf "Congratulation !! This Year Carbon Emission
            Achievement is %d TonCO2e " (valccemission*targeterfvalue)
logInfo @String $ printf "Carbon Credit Rewards will be creditted
            to Account Beneficiary if Validated Carbon Emission Target
            Year below Science Based Target Value "
where
      hasCorrectDatum :: ChainIndexTxOut -> Bool
      hasCorrectDatum (ScriptChainIndexTxOut _ _ (Right(Datum datum)) _)=
          case PlutusTx.fromBuiltinData datum of
      Just d  ->  (valccemission * targeterfvalue) ==
                  (set_SBTTonCO2E d) * (set_ERF d) &&
                  pin_validation == (set_pin d)
      Nothing -> False
      hasCorrectDatum _ = False
```

Source Code 8 Science Based Target Carbon Emission Contract Validation End Point Off-Chain

- Science Based Target Carbon Emission Contract Reward End Point Off-Chain

This Smart Contract End Point is consolidated all the Smart Contract which include Contract Script and Reward Schema.

```
-- Carbon Credit Science Base Target Rewards endpoints.
endpoints :: AsContractError e => Contract () CCRewardsSchema e ()
endpoints = awaitPromise (ccsbtrewards' `select` carbonemissionupdate')
      >> endpoints
      where
            ccsbtrewards' = endpoint @"Science Base Target GHG Data
                  Center and Funds Deposit" ccsbtrewards
            carbonemissionupdate' = endpoint @"Updated Current GHG Data
                  Center and Released Funds Deposit" carbonemissionupdate
mkSchemaDefinitions ''CCRewardsSchema
```

Source Code 9 Science Based Target Carbon Emission Contract Reward End Point Off-Chain

2.2 Net Zero Carbon Campaign Smart Contract Version 2.0 (Bounty Auction)

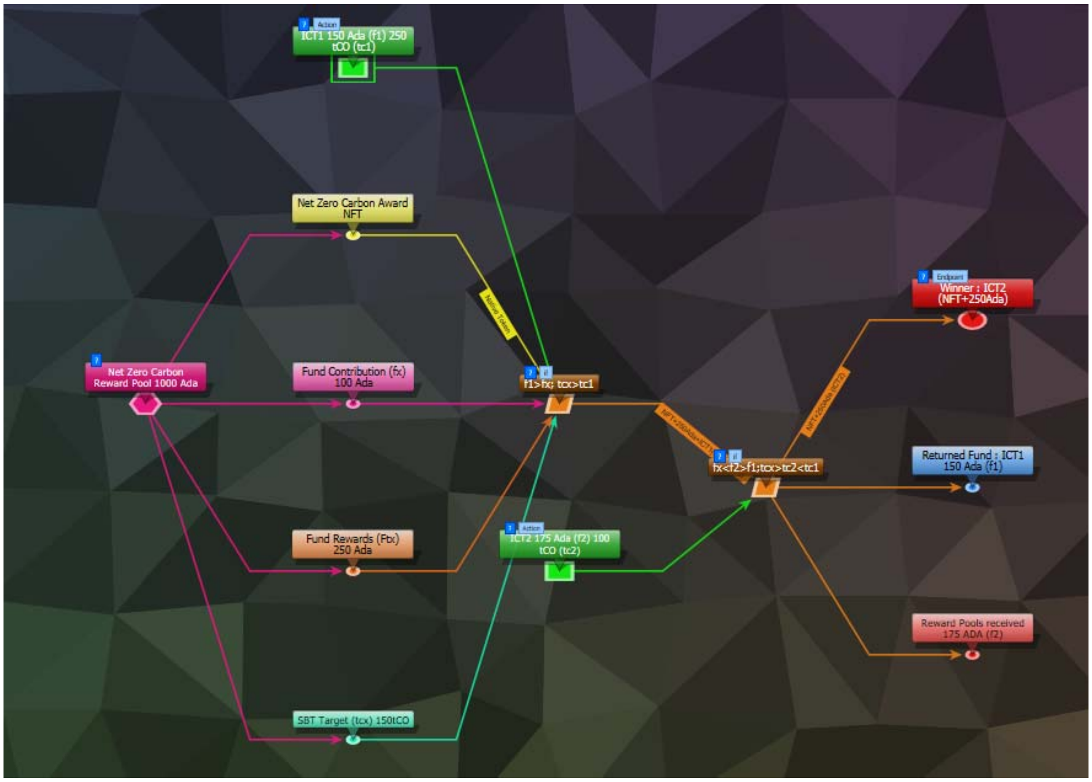

Figure 8 Net Zero Carbon Campaign Smart Contract Version 2 (Bounty Auction)

The features logic algorithm of Science Based Target Net Zero Carbon Emission Campaign as part of algorithm upgraded, it has been developed with idea of Bounty and Auction Campaign. The participant will try to achieve the global target achievement, meanwhile the participant will ensure their report submission through report warranty scheme fund which part of fund contribution. The Smart Contract algorithm defined as:

2.2.1 On-chain Smart Contract, which consists of:

- Carbon Credit SBT Parameter Input Data and Validation Report Parameter (On-Chain)

On-Chain Input Data defined as input parameter which segregated as Science Based Target Campaign Publisher (SBTCampaign) and Science Based Target Campaign Participant (CampaignSBT).

Science Based Target Campaign Publisher (SBTCampaign) consists of Publisher Public Hash Key Beneficiary Address (aPublisher), Campaign Duration (aChampaignDuration), Minimum Fund Contribution (aMinContribution), Maximum Target Emission in tCO2 (aMaxCO2), Currency (aCurrency), Reward Token (aToken).

```
-- Net Zero Carbon Champaign Rewards Publisher Instance Data
data SBTChampaign = SBTChampaign
 { aPublisher            :: !PaymentPubKeyHash
 , aChampainDuration     :: !POSIXTime
 , aMinContribution      :: !Integer
 , aMaxCO2               :: !Integer
 , aCurrency             :: !CurrencySymbol
 , aToken                :: !TokenName
 } deriving (P.Show, Generic, ToJSON, FromJSON, ToSchema)

instance Eq SBTChampaign where
   {-# INLINABLE (==) #-}
   a == b = (aPublisher          a == aPublisher         b) &&
            (aChampainDuration     a == aChampainDuration  b) &&
            (aMinContribution      a == aMinContribution   b) &&
             (aMaxCO2               a == aMaxCO2            b) &&
             (aCurrency             a == aCurrency          b) &&
             (aToken                a == aToken             b)
PlutusTx.unstableMakeIsData ''SBTChampaign
PlutusTx.makeLift ''SBTChampaign

-- Net Zero Carbon Champaign Rewards Participant Instance Data
data ChampaignSBT = ChampaignSBT
{ bParticipant          :: !PaymentPubKeyHash
, bContribution         :: !Integer
, bCO2Emission          :: !Integer
} deriving P.Show

instance Eq ChampaignSBT where
 {-# INLINABLE (==) #-}
      b == c = (bParticipant          b == bParticipant  c) &&
               (bContribution         b == bContribution c) &&
               (bCO2Emission          b == bCO2Emission  c)
PlutusTx.unstableMakeIsData ''ChampaignSBT
PlutusTx.makeLift ''ChampaignSBT
```

```
-- Net Zero Carbon Champaign Rewards Champaign Action Data
data ChampaignAction = MkChampaign ChampaignSBT | Close
      deriving P.Show
PlutusTx.unstableMakeIsData ''ChampaignAction
PlutusTx.makeLift ''ChampaignAction
-- Wrapped Net Zero Carbon Champaign Rewards Datum Data
data SBTChampaignDatum = SBTChampaignDatum
      -- Wrapped SBT Champaign Publisher Data : aPublisher,
      aChampainDuration, aMinContribution, aMaxCO2, aCurrency, aToken
      { adChampaign           :: !SBTChampaign
      -- Wrapped SBT Champaign Participant Data : bParticipant,
      bContribution, bCO2Emission
      , adHighestContribution :: !(Maybe ChampaignSBT)
      } deriving P.Show

PlutusTx.unstableMakeIsData ''SBTChampaignDatum
PlutusTx.makeLift ''SBTChampaignDatum

data Champaigned
instance Scripts.ValidatorTypes Champaigned where
      type instance RedeemerType Champaigned = ChampaignAction
      type instance DatumType Champaigned = SBTChampaignDatum
```

Source Code 10 Carbon Credit SBT Parameter Input Data and Validation Report Parameter (On-Chain)

- Script for Minimum Contribution and Max CO2 Emission

The Minimum Contribution algorithm is to ensure that participants will submit their contribution always above the set minimum contribution and as warranty of their report submission. The highest contribution will be relevant with higher in transparency, auditable, and trust of the report submission.

Meanwhile the Max CO2 Emission Script is part of the validation of the emission report which has been submitted by participant whether meet the global target set achievement.

```
{-# INLINABLE minContribute #-}
minContribute :: SBTChampaignDatum -> Integer
minContribute SBTChampaignDatum{..} = case adHighestContribution of
    Nothing       ->  aMinContribution  adChampaign
    Just ChampaignSBT{..} ->   bContribution + 1
```

```
{-# INLINABLE maxCO2Emission #-}
maxCO2Emission :: SBTChampaignDatum -> Integer
maxCO2Emission SBTChampaignDatum{..} = case adHighestContribution of
    Nothing      ->  aMaxCO2  adChampaign
    Just ChampaignSBT{..} ->   bCO2Emission
```

Source Code 11 Script for Minimum Contribution and Max CO2 Emission

- Carbon Credit SBT Validation Smart Contract On Chain

Carbon Credit SBT Validation Smart Contract On-Chain defined as Smart Contract which run on On-Chain. The program consists of Champaign Input Datum, Champaign Action, and Script Context. This contract defined all the setup parameter locked datum prior released the rewards, Hash Log Info Error On-Chain, and sub routine script as part of Champaign Validator Script.

```
-- On-Chain Validator Smart Contract
{-# INLINABLE mkChampaignValidator #-}
mkChampaignValidator :: SBTChampaignDatum -> ChampaignAction ->
 ScriptContext -> Bool
mkChampaignValidator ad redeemer ctx =
    traceIfFalse "wrong input value" correctInputValue &&
    case redeemer of
        MkChampaign b@ChampaignSBT{..} ->
           traceIfFalse "Contribution Funds too low"
                  (sufficientBid bContribution)         &&
           traceIfFalse "wrong output datum" (correctBidOutputDatum b)  &&
           traceIfFalse "wrong output value" (correctBidOutputValue
             bContribution) &&
           traceIfFalse "wrong refund"       correctBidRefund           &&
           traceIfFalse "too late"           correctBidSlotRange
        Close           ->
            traceIfFalse "too early" correctCloseSlotRange &&
            case adHighestContribution ad of
                Nothing      ->
                    traceIfFalse "expected seller to get token"
                         (getsValue (aPublisher auction) $ tokenValue <>
                         Ada.lovelaceValueOf minLovelace)
                Just ChampaignSBT{..} ->
                    traceIfFalse "expected highest Participant to get
```

```
                   token" (getsValue bParticipant  $ tokenValue <> Ada
             .lovelaceValueOf minLovelace) &&
              traceIfFalse "expected seller to get highest
                                Contributor" (getsValue (aPublisher
                                auction) $ Ada.lovelaceValueOf
                                bContribution)
             where
                   info :: TxInfo
                   info = scriptContextTxInfo ctx
                   input :: TxInInfo
                   input =
                         let
                         isScriptInput i = case (txOutDatumHash .
                               txInInfoResolved) i of
                         Nothing -> False
                   Just _  -> True
             xs = [i | i <- txInfoInputs info, isScriptInput i]
             in
             case xs of
             [i] -> i
              _   -> traceError "expected exactly one script input"

        inVal :: Value
        inVal = txOutValue . txInInfoResolved $ input
        auction :: SBTChampaign
        auction = adChampaign ad
        tokenValue :: Value
        tokenValue = Value.singleton (aCurrency auction) (aToken auction) 1
        correctInputValue :: Bool
        correctInputValue = inVal == case adHighestContribution ad of
              Nothing      -> tokenValue <> Ada.lovelaceValueOf minLovelace
               Just ChampaignSBT{..} -> tokenValue <>
                           Ada.lovelaceValueOf (minLovelace + bContribution)
        sufficientBid :: Integer -> Bool
        sufficientBid amount = amount >= minContribute ad
        ownOutput   :: TxOut
        outputDatum :: SBTChampaignDatum
        (ownOutput, outputDatum) = case getContinuingOutputs ctx of
          [o] -> case txOutDatumHash o of
              Nothing   -> traceError "wrong output type"
              Just h -> case findDatum h info of
                  Nothing        -> traceError "datum not found"
                  Just (Datum d) ->  case PlutusTx.fromBuiltinData d of
```

```
                     Just ad' -> (o, ad')
                     Nothing  -> traceError "error decoding data"
             _   -> traceError "expected exactly one continuing output"
        correctBidOutputDatum :: ChampaignSBT -> Bool
        correctBidOutputDatum b = (adChampaign outputDatum == auction)   &&
                                  (adHighestContribution outputDatum == Just b)
        correctBidOutputValue :: Integer -> Bool
        correctBidOutputValue amount =
            txOutValue ownOutput == tokenValue <> Ada.lovelaceValueOf
                (minLovelace + amount)
        correctBidRefund :: Bool
        correctBidRefund = case adHighestContribution ad of
            Nothing      -> True
            Just ChampaignSBT{..} ->
              let
                os = [ o
                     | o <- txInfoOutputs info
                     , txOutAddress o == pubKeyHashAddress bParticipant
                           Nothing
                     ]
              in
                case os of
                    [o] -> txOutValue o == Ada.lovelaceValueOf bContribution
                    _   -> traceError "expected exactly one refund output"
        correctBidSlotRange :: Bool
        correctBidSlotRange = to (aChampainDuration auction) `contains`
                                   txInfoValidRange info
        correctCloseSlotRange :: Bool
        correctCloseSlotRange = from (aChampainDuration auction) `contains`
          txInfoValidRange info
        getsValue :: PaymentPubKeyHash -> Value -> Bool
        getsValue h v =
          let
            [o] = [ o'
                  | o' <- txInfoOutputs info
                  , txOutValue o' == v
                  ]
          in
            txOutAddress o == pubKeyHashAddress h Nothing

typedAuctionValidator :: Scripts.TypedValidator Champaigned
typedAuctionValidator = Scripts.mkTypedValidator @Champaigned
    $$(PlutusTx.compile [|| mkChampaignValidator ||])
```

```
    $$(PlutusTx.compile [|| wrap ||])
  where
    wrap = Scripts.wrapValidator @SBTChampaignDatum @ChampaignAction

sbtchampaignValidator :: Validator
sbtchampaignValidator = Scripts.validatorScript typedAuctionValidator
auctionHash :: Ledger.ValidatorHash
auctionHash = Scripts.validatorHash typedAuctionValidator
auctionAddress :: Ledger.Address
auctionAddress = scriptHashAddress auctionHash
```

Source Code 12 Carbon Credit SBT Validation Smart Contract On Chain

2.2.2 Off-chain Smart Contract, which consists of:

- Carbon Credit SBT Parameter Input Data and Validation Report Parameter (Off-Chain)

Off-Chain Input Data is defined as input parameter which segregated as Campaign Start (StartParams), Bidding (BidParams), Validation (ValidationParams), and Campaign Closed Parameters (CloseParams).

Campaign Start Parameter (StartParams) input data consists of campaign duration (setChampaignDuration), minimum contribution (setMinFundContribution), maximum target CO2 emission in tCO2 (setMaxCO2EmissionTarget), currency (setCurrency), and reward token (setRewardTokenName).

Bidding Parameter (BidParams) input data consists of Campaign Currency (champaignCurrency), Campaign Generated Native Reward Token (champaignRewardsToken), CO2 submission Report (submissionCO2EmissionReport) and Fund Contribution (champaignFundsContribution).

Validation Parameter (ValidationParams) input data consists of validated currency (valCurrency), validated Token (valToken), and validated CO2 Emission (valCO2Emission)

Champaign Closed Parameter (CloseParams) input data consist of currency (cpCurrency) and token (cpToken) information.

```
-- Off Chain Input Data Script Parameters
data StartParams = StartParams
    { setChampaignDuration          :: !POSIXTime
    , setMinFundContribution        :: !Integer
    , setMaxCO2EmissionTarget       :: !Integer
    , setCurrency                   :: !CurrencySymbol
    , setRewardTokenName            :: !TokenName
    } deriving (Generic, ToJSON, FromJSON, ToSchema)


data BidParams = BidParams
    { champaignCurrency             :: !CurrencySymbol
    , champaignRewardsToken         :: !TokenName
    , champignFundsContribution     :: !Integer
    , submissionCOEmissionReport    :: !Integer
    } deriving (Generic, ToJSON, FromJSON, ToSchema)

data ValidationParams = ValidationParams
    { valCurrency       :: !CurrencySymbol
    , valToken          :: !TokenName
    , valCO2Emission    :: !Integer
    } deriving (Generic, ToJSON, FromJSON, ToSchema)

data CloseParams = CloseParams
    { cpCurrency :: !CurrencySymbol
    , cpToken    :: !TokenName
    } deriving (Generic, ToJSON, FromJSON, ToSchema)
```

Source Code 13 Carbon Credit SBT Parameter Input Data and Validation Report Parameter (Off-Chain)

- Carbon Credit SBT Schema (Off-Chain)

Carbon Credit SBT Schema is to tie up all the code in Off Chain Smart Contract End Points such Campaign Start, Campaign Auction, and Campaign Close in one endpoint.

```
-- Off Chain Campaign Schema
type SBTChampaignSchema =
        Endpoint "Net Zero Carbon Champaign Started"        StartParams
    .\/ Endpoint "Net Zero Carbon Champaign Submission"     BidParams
    .\/ Endpoint "Net Zero Carbon Champaign Ended"          CloseParams
```

Source Code 14 Carbon Credit SBT Schema (Off-Chain)

- Carbon Credit SBT Champaign Started End Point (Off-Chain)

Carbon Credit SBT Campaign Started End Points defined as Smart Contract which run on Off-Chain where the program consists of consolidated Smart Contract Error, Champaign Start Input Data, and Contract Script Context. This contract defined all the setup parameter locked datum prior released the rewards, Hash Log Info Error On-Chain, and sub routine script as part of Champaign Validator Script.

```
-- Off Chain Campaign Started End Points
champaignstarted :: AsContractError e => StartParams -> Contract w s e ()
champaignstarted StartParams{..} = do
    pkh <- ownPaymentPubKeyHash
    let a = SBTChampaign
                { aPublisher          = pkh
                , aChampainDuration   = setChampaignDuration
                , aMinContribution    = setMinFundContribution
                , aMaxCO2             =  setMaxCO2EmissionTarget
                , aCurrency           = setCurrency
                , aToken              = setRewardTokenName
                }
        d = SBTChampaignDatum
                { adChampaign    = a
                , adHighestContribution = Nothing
                }
        v = Value.singleton setCurrency setRewardTokenName 1 <>
            Ada.lovelaceValueOf minLovelace
            tx = Constraints.mustPayToTheScript d v
            ledgerTx <- submitTxConstraints typedAuctionValidator tx
            void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
            logInfo @P.String $ printf "started SBT Champaign %s for
                  token %s" (P.show a) (P.show v)
```

Source Code 15 Carbon Credit SBT Champaign Started End Point (Off-Chain)

- Carbon Credit SBT Champaign Bidding/ Auction Submission End Point (Off-Chain)

Carbon Credit SBT Campaign Started End Points defined as Smart Contract which run on Off-Chain where the program consists of consolidated Bidding Auction Input Parameters and Contract Script Context. This contract defined all the setup parameters to unlock or release the rewards through the highest contributor and lowest CO2 emission achiever, Hash Log Info Error On-Chain, and sub routine script as part of Champaign Validator Script.

In this section we utilized cascade log error tracing method to make easier for publisher and participant to trace the auction which led to transparency of auction.

```
-- Off Chain Campaign Bidding Submission End Points
champaign :: forall w s. BidParams -> Contract w s Text ()
champaign BidParams{..} = do
      (oref, o, d@SBTChampaignDatum{..}) <- findCampaign
            champaignCurrency champaignRewardsToken
      logInfo @P.String $ printf "found SBT Champaign utxo
            with datum %s" (P.show d)

-- Error Log and Unlocked Rewards Schema
  if (champignFundsContribution  < minContribute d)
    then logError @P.String $ printf "Contribution Funds lower than
      minimal Contribution Required as %d" (minContribute d)
    else if (submissionCOEmissionReport > maxCO2Emission d)
        then logError @P.String $ printf "CO2 Emission higher than
            maximum allowed in the Campaign as %d tCO2/Year"
            (maxCO2Emission d)
    else do
        pkh <- ownPaymentPubKeyHash
        let b  = ChampaignSBT {bParticipant  = pkh, bContribution =
                   champignFundsContribution, bCO2Emission =
                   submissionCOEmissionReport}
            d' = d {adHighestContribution = Just b}
            v  = Value.singleton champaignCurrency champaignRewardsToken 1
                    <> Ada.lovelaceValueOf (minLovelace +
                         champignFundsContribution )
            r  = Redeemer $ PlutusTx.toBuiltinData $ MkChampaign b

            lookups = Constraints.typedValidatorLookups
                         typedAuctionValidator P.<>
                    Constraints.otherScript sbtchampaignValidator
                         P.<>
                    Constraints.unspentOutputs (Map.singleton oref o)
            tx      = case adHighestContribution of
                        Nothing ->
                               Constraints.mustPayToTheScript d' v     <>
```

```
                                Constraints.mustValidateIn (to
                                      $ aChampainDuration adChampaign)  <>
                                Constraints.mustSpendScriptOutput oref r
                         Just ChampaignSBT{..} ->
                                Constraints.mustPayToTheScript d' v      <>

                                Constraints.mustPayToPubKey bParticipant
                                     (Ada.lovelaceValueOf bContribution) <>
                                Constraints.mustValidateIn (to $
                                      aChampainDuration adChampaign)     <>
                                Constraints.mustSpendScriptOutput oref r
             ledgerTx <- submitTxConstraintsWith lookups tx
             void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
             logInfo @P.String $ printf "Made Net Zero Champaign Funds
                   Available of %d lovelace in Champaigned %s for Rewards
                   Token (%s, %s)"
             champignFundsContribution
             (P.show adChampaign)
             (P.show champaignCurrency)
             (P.show champaignRewardsToken)
```

Source Code 16 Carbon Credit SBT Champaign Bidding/ Auction Submission End Point (Off-Chain)

- Carbon Credit SBT Champaign Ended End Point (Off-Chain)

Carbon Credit SBT Campaign Ended defined as Smart Contract which run on Off-Chain as part the action to end the campaign when reached the end of campaign duration. The program consists of consolidated to Close the Campaign, and Contract Script Context. This contract defined all the setup parameters required to close the campaign which included closed script and log error information.

```
-- Champaign Ended Contract
champaignended :: forall w s. CloseParams -> Contract w s Text ()
champaignended CloseParams{..} = do
    (oref, o, d@SBTChampaignDatum{..}) <- findCampaign cpCurrency cpToken
    logInfo @P.String $ printf "Found Champaign utxo with datum %s"
      (P.show d)

    let t      = Value.singleton cpCurrency cpToken 1
        r      = Redeemer $ PlutusTx.toBuiltinData Close
        seller = aPublisher adChampaign

        lookups = Constraints.typedValidatorLookups typedAuctionValidator
```

```
                    P.<>
                  Constraints.otherScript sbtchampaignValidator     P.<>
                  Constraints.unspentOutputs (Map.singleton oref o)
        tx      = case adHighestContribution of
                    Nothing      ->
                         Constraints.mustPayToPubKey seller (t   <>
                         Ada.lovelaceValueOf minLovelace)        <>
                         Constraints.mustValidateIn (from $
                               aChampainDuration adChampaign)    <>
                         Constraints.mustSpendScriptOutput oref r
                    Just ChampaignSBT{..} ->
                         Constraints.mustPayToPubKey bParticipant  (t <>
                         Ada.lovelaceValueOf minLovelace)             <>
                         Constraints.mustPayToPubKey seller
                               (Ada.lovelaceValueOf bContribution)    <>
                         Constraints.mustValidateIn (from $
                               aChampainDuration adChampaign)         <>
                         Constraints.mustSpendScriptOutput oref r
    ledgerTx <- submitTxConstraintsWith lookups tx
    void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
    logInfo @P.String $ printf "Closed SBT Champaign %s for token (%s, %s)"
        (P.show adChampaign)
        (P.show cpCurrency)
        (P.show cpToken)

findCampaign :: CurrencySymbol
            -> TokenName
            -> Contract w s Text
                   (TxOutRef, ChainIndexTxOut, SBTChampaignDatum)
findCampaign cs tn = do
    utxos <- utxosAt $ scriptHashAddress auctionHash
    let xs = [ (oref, o)
             | (oref, o) <- Map.toList utxos
             , Value.valueOf (_ciTxOutValue o) cs tn == 1
             ]
    case xs of
        [(oref, o)] -> case _ciTxOutDatum o of
            Left _          -> throwError "Datum missing"
            Right (Datum e) -> case PlutusTx.fromBuiltinData e of
                Nothing -> throwError "Datum has wrong type"
                Just d@SBTChampaignDatum{..}
                    | aCurrency adChampaign == cs && aToken adChampaign == t
n -> return (oref, o, d)
```

```
                        | otherwise ->
                            throwError "Champaigned token missmatch"
            _           -> throwError "Champaigned utxo not found"
```

Source Code 17 Carbon Credit SBT Champaign Ended End Point (Off-Chain)

- Carbon Credit SBT Champaign Ended Smart Contract End Point (Off-Chain)

This Smart Contract End Point is consolidated all the Smart Contract which include Contract Script and Reward Schema.

```
-- Off Chain End Points Contract Schema
endpoints :: Contract () SBTChampaignSchema Text ()
endpoints = awaitPromise (champaignstarted' `select` champaign' `select`
            champaignended') >> endpoints
  where
    champaignstarted'   = endpoint @"Net Zero Carbon Champaign Started"
                          champaignstarted
    champaign'          = endpoint @"Net Zero Carbon Champaign Submission"
                          champaign
    champaignended'     = endpoint @"Net Zero Carbon Champaign Ended"
                          champaignended
mkSchemaDefinitions ''SBTChampaignSchema
myToken :: KnownCurrency
myToken = KnownCurrency (ValidatorHash "f") "Token"
          (TokenName "Net Zero Carbon Awards" :| [])
mkKnownCurrencies ['myToken]
```

Source Code 18 Carbon Credit SBT Champaign Ended Smart Contract End Point (Off-Chain)

3. Net Zero Carbon Campaign Smart Contract – Smart Contract Interaction

Smart Contract interaction as part of simulation test through compile and deploy the Plutus Background Application. Several step to perform the test as below:

3.1 Net Zero Carbon Campaign Genesis Smart Contract Simulation Setup Parameter (3 Wallet)

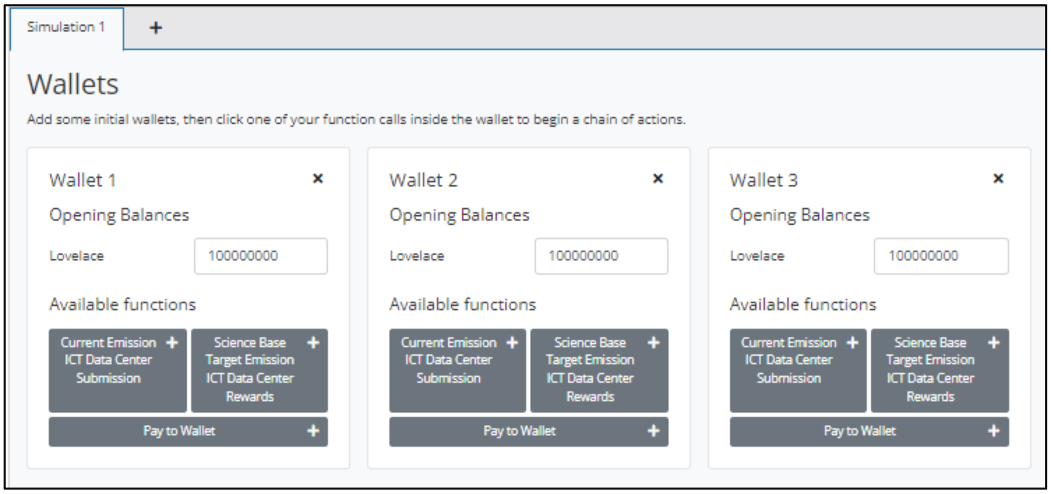

Figure 9 Smart Contract Simulation Using 3 (Three) Wallet with Different Conditions

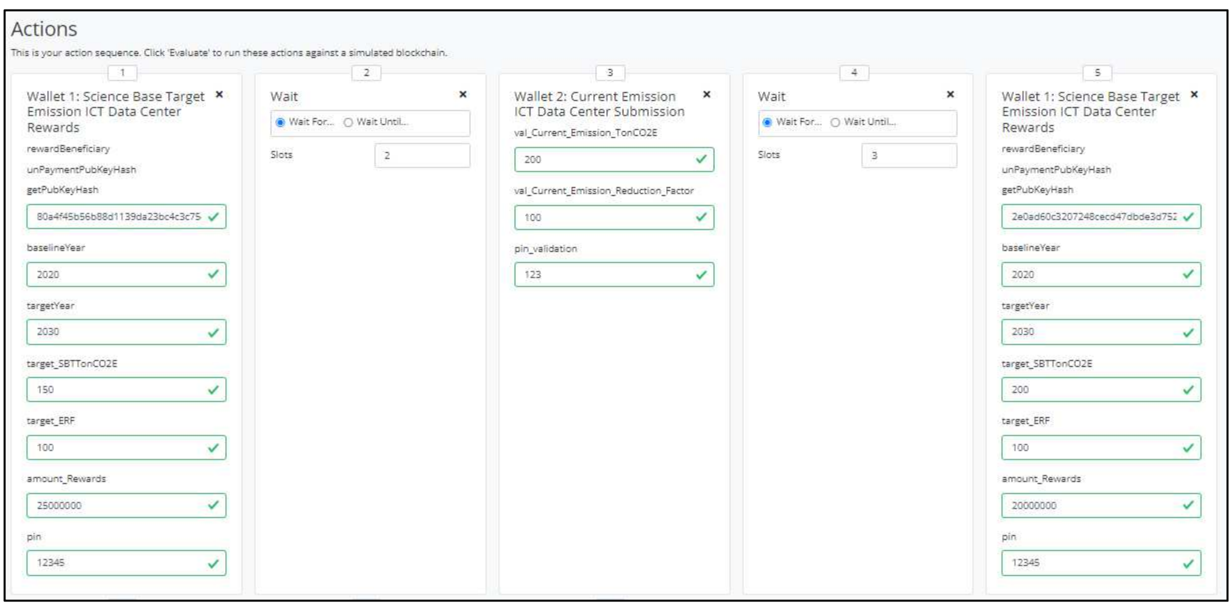

Figure 10 Smart Contract Simulation Using 3 (Three) Wallet with Different Conditions

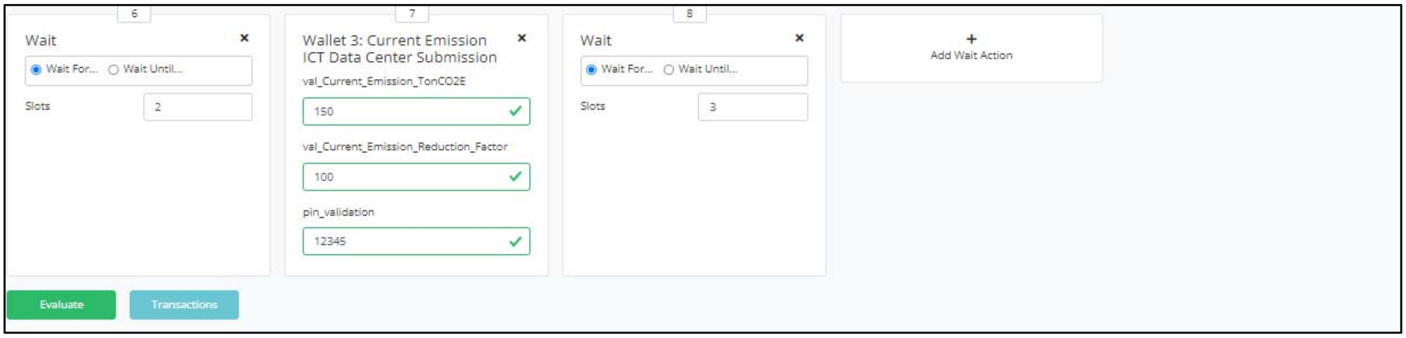

Figure 11 Smart Contract Simulation Using 3 (Three) Wallet with Different Conditions

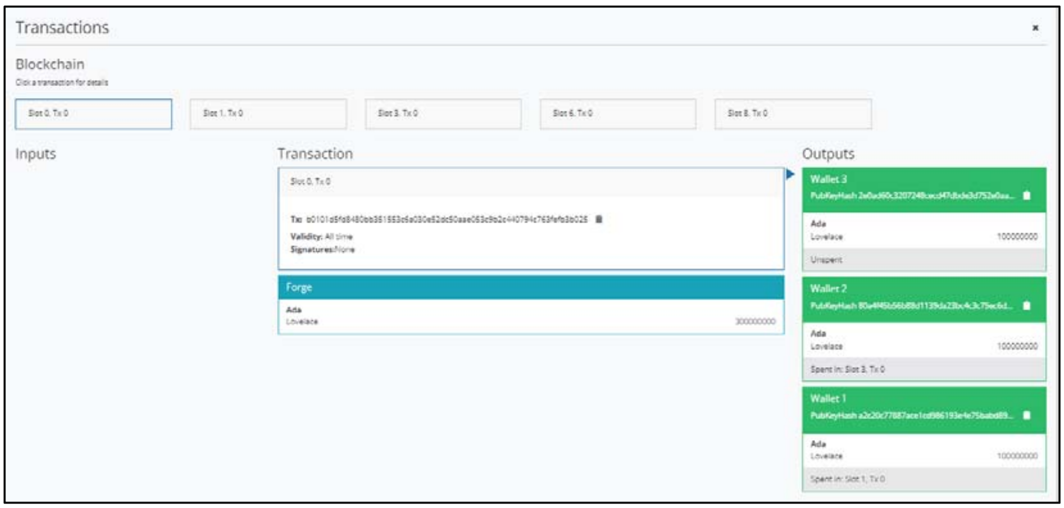

Figure 12 Smart Contract Simulation Result On Slot 0 Tx 0

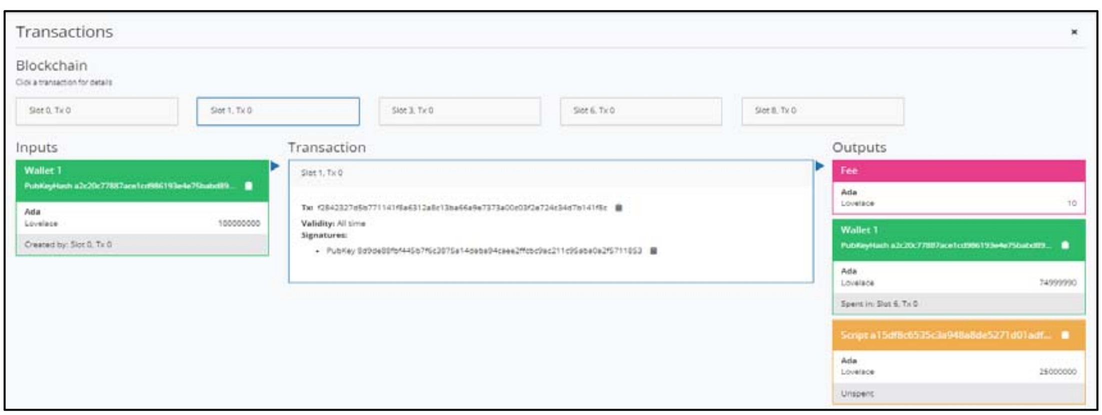

Figure 13 Smart Contract Simulation Result On Slot 1 Tx 0

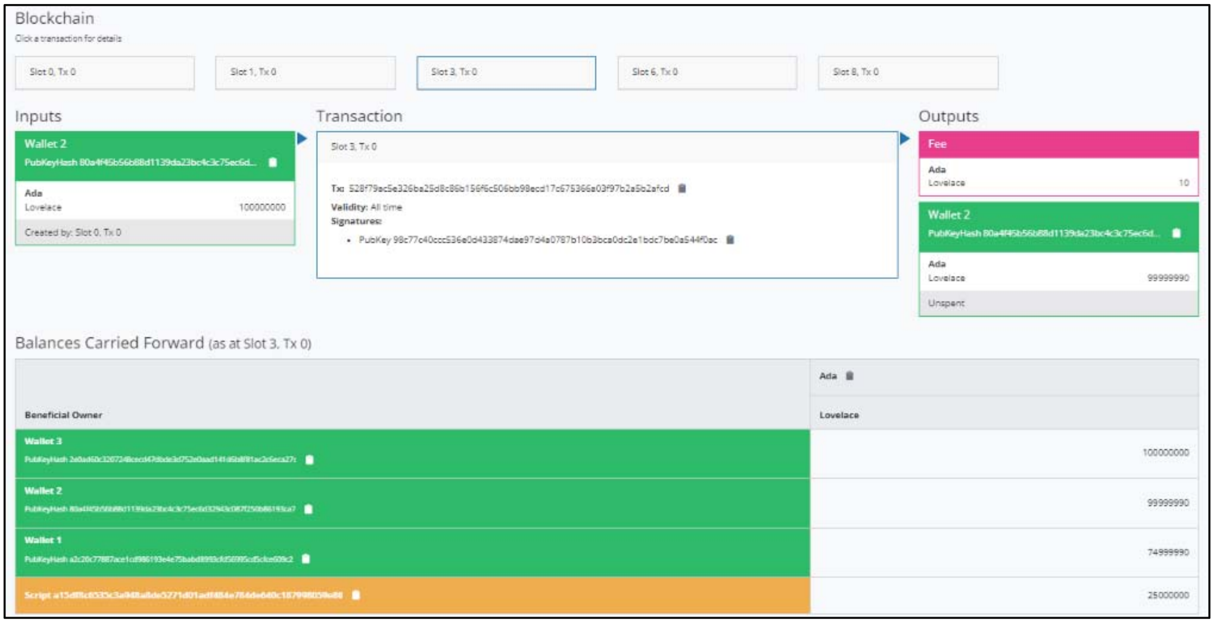

Figure 14 Smart Contract Simulation Result On Slot 3 Tx 0

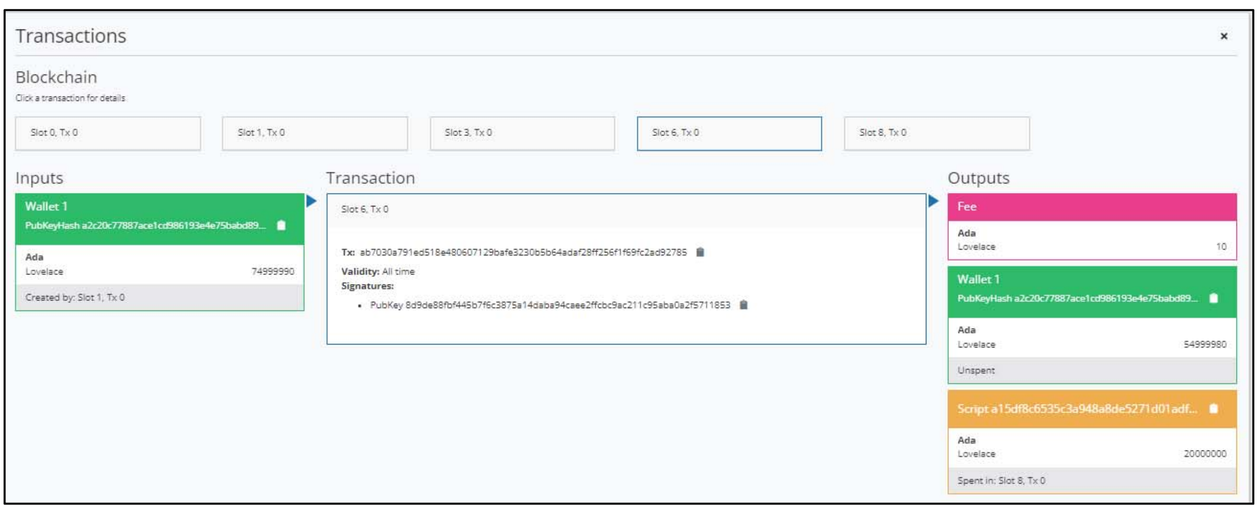

Figure 15 Smart Contract Simulation Result On Slot 6 Tx 0

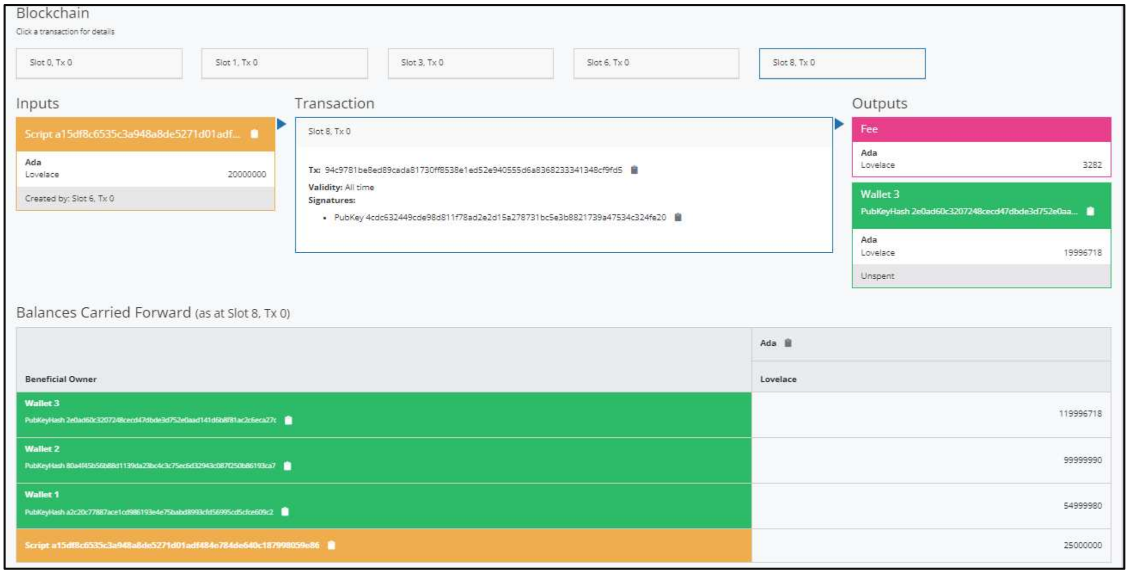

Figure 16 Smart Contract Simulation Result On Slot 8 Tx 0

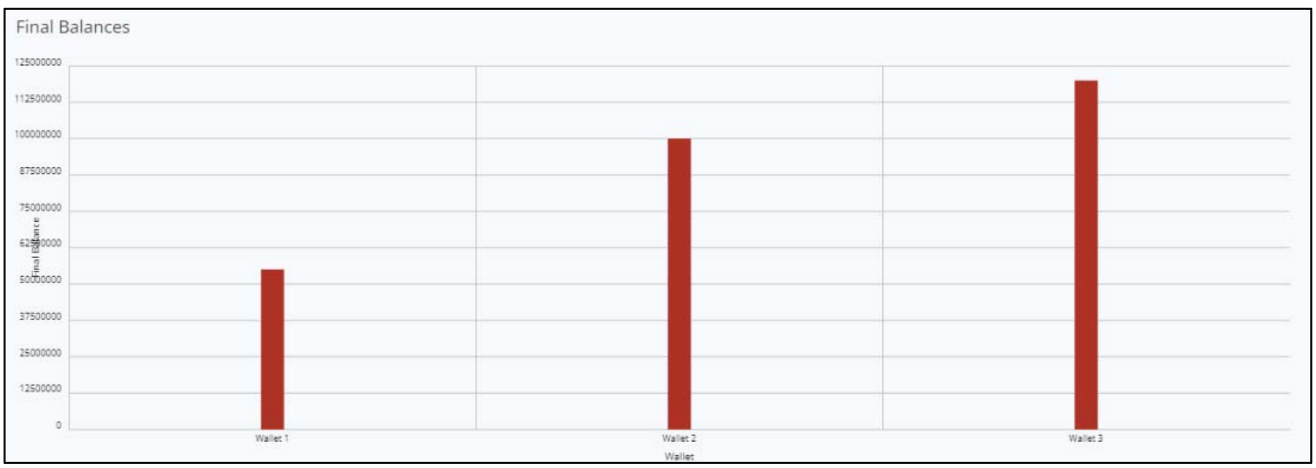

Figure 17 Smart Contract Simulation Result (Overall Final Wallet Balance)

Figure 18 Smart Contract Simulation Result (Transaction Info Logs for Wallet2 not meet Target)

Figure 19 Smart Contract Simulation Result (Transaction Info Logs for Wallet3 Meet Target)

3.2 Net Zero Carbon Campaign Smart Contract Version 2 Simulation Setup (Bounty Auction)

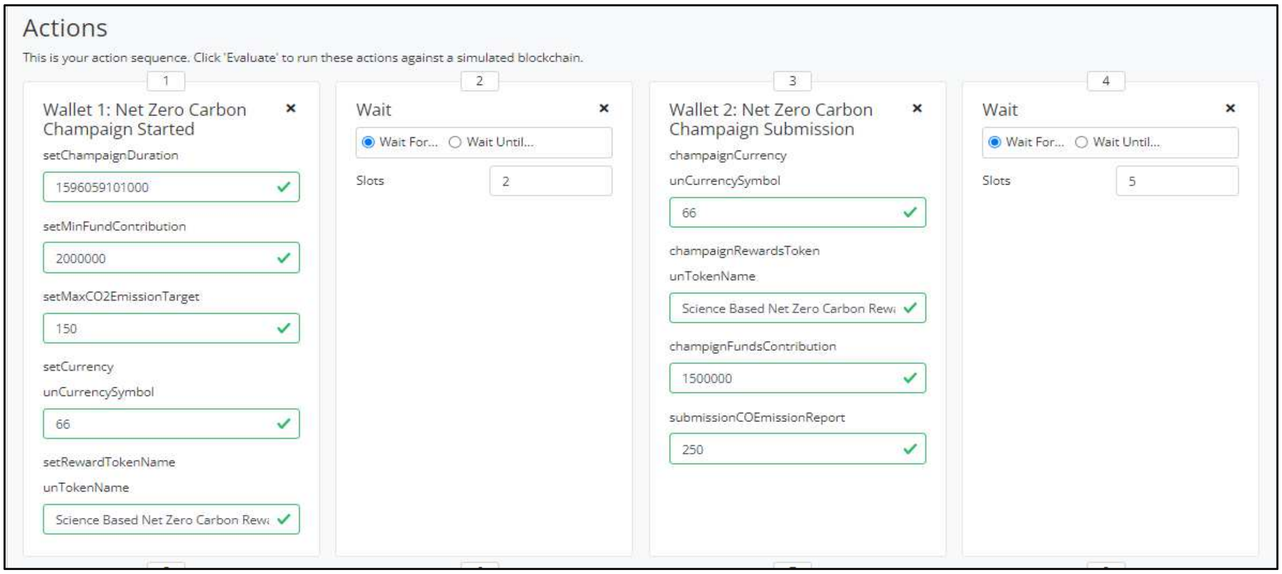

Figure 20 Smart Contract Simulation Using 3 (Three) Wallet with Different Conditions

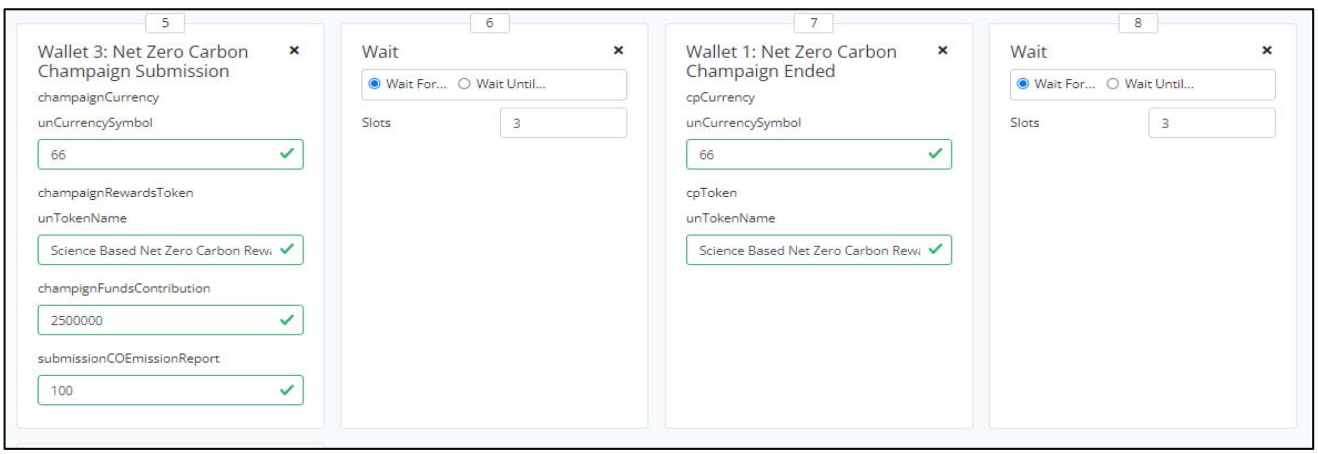

Figure 21 Smart Contract Simulation Using 3 (Three) Wallet with Different Conditions

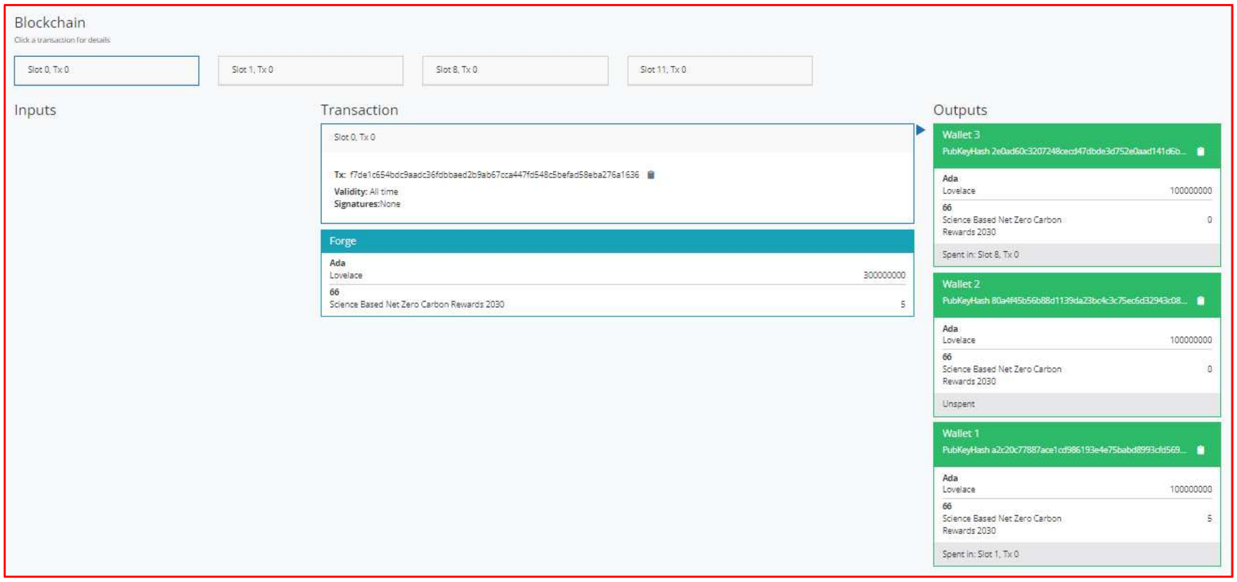

Figure 22 Smart Contract Simulation Result On Slot 0 Tx 0

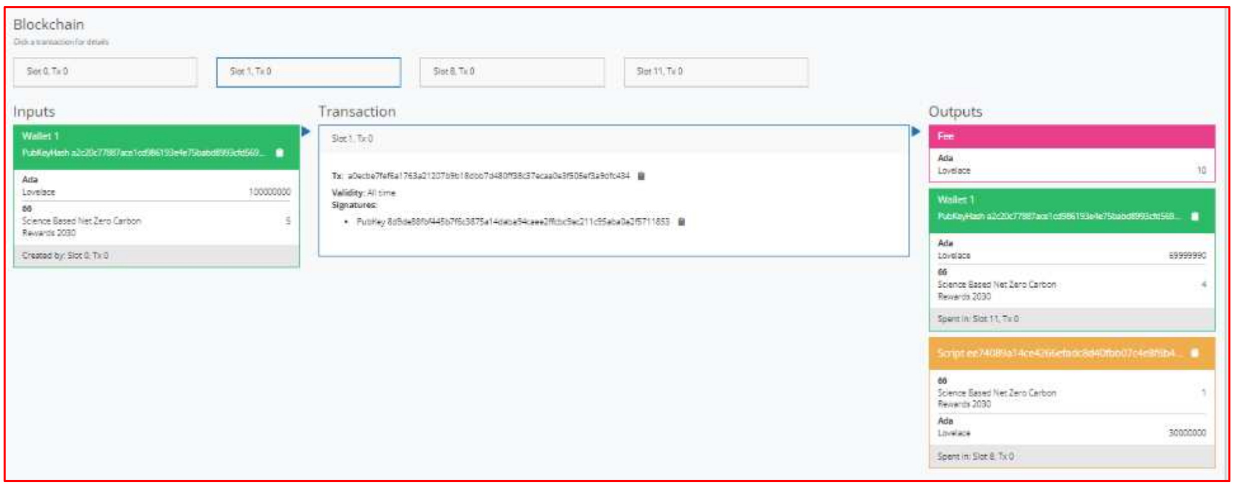

Figure 23 Smart Contract Simulation Result On Slot 1 Tx 0

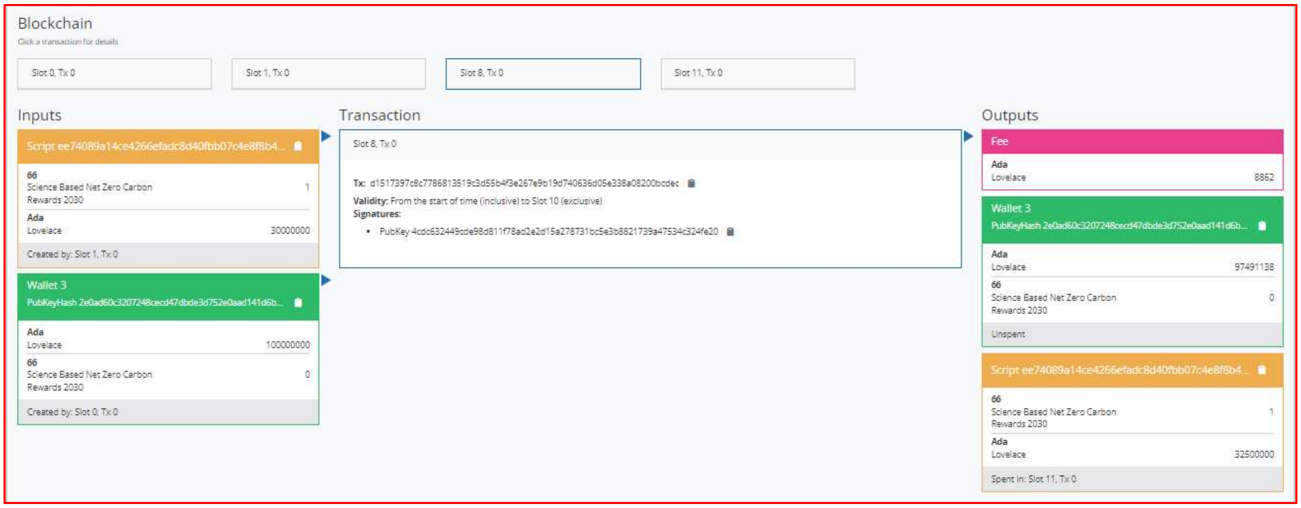

Figure 24 Smart Contract Simulation Result On Slot 8 Tx 0

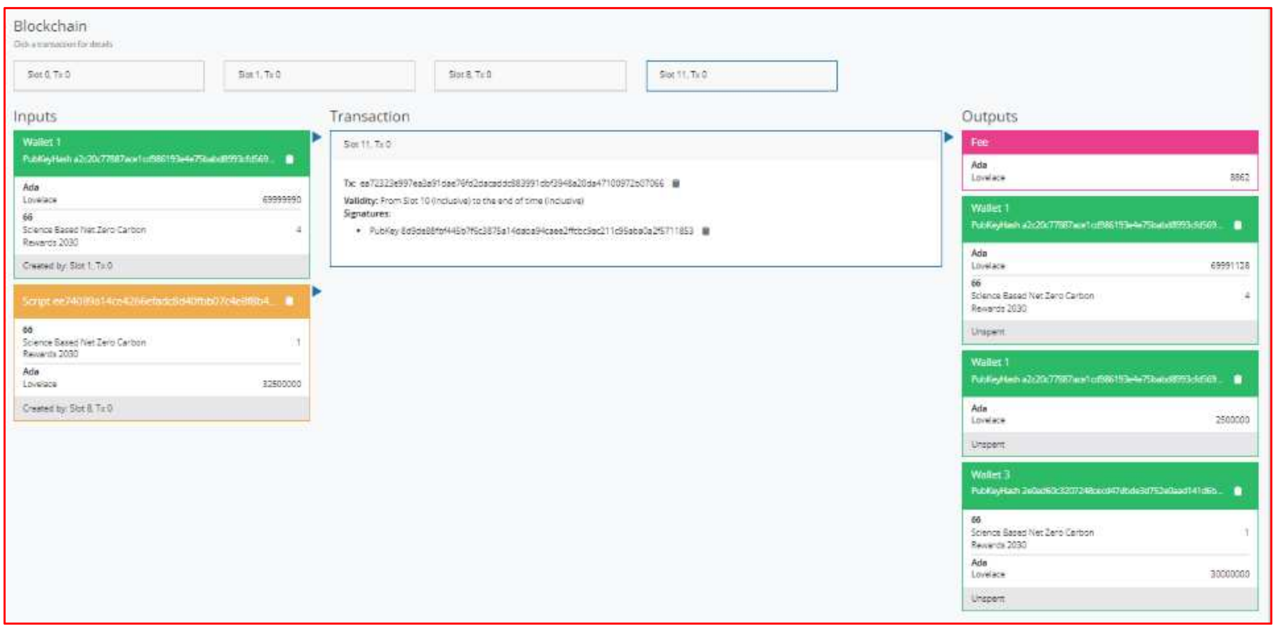

Figure 25 Smart Contract Simulation Result On Slot 11 Tx 0

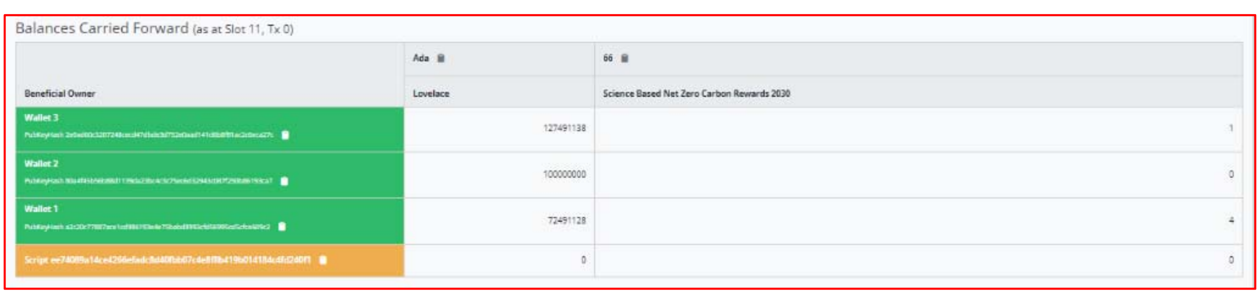

Figure 26 Smart Contract Simulation Result (Overall Final Balance - Fund and Token)

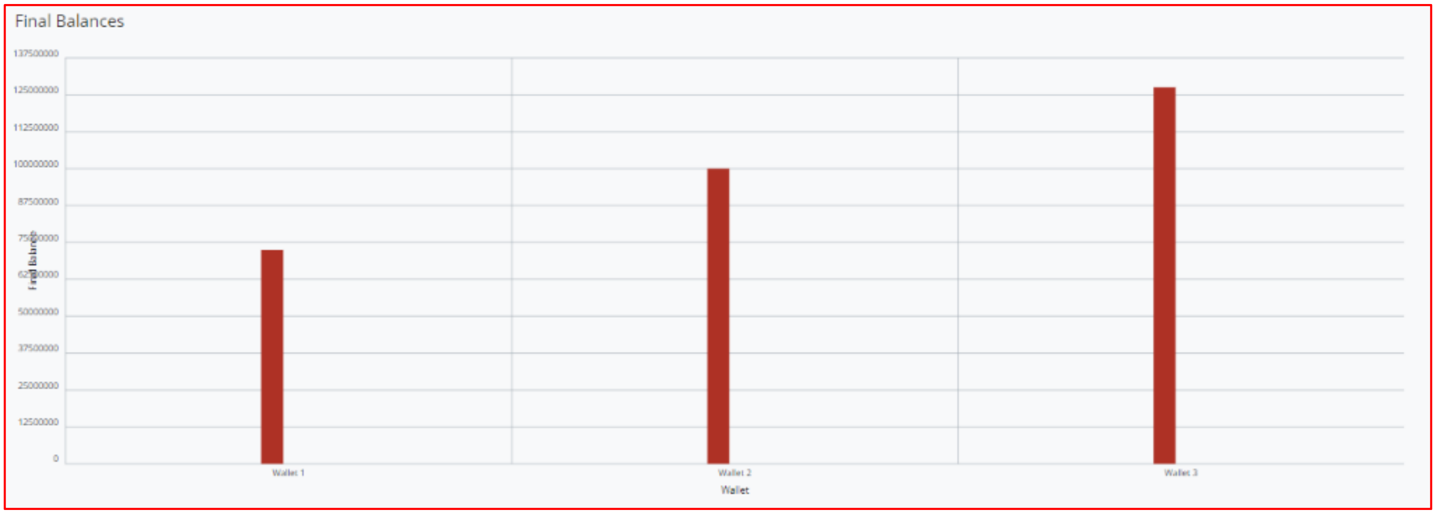

Figure 27 Smart Contract Simulation Result (Overall Final Wallet Balance - Fund)

Figure 28 Smart Contract Simulation Log Info for Wallet 2 on Slot 3

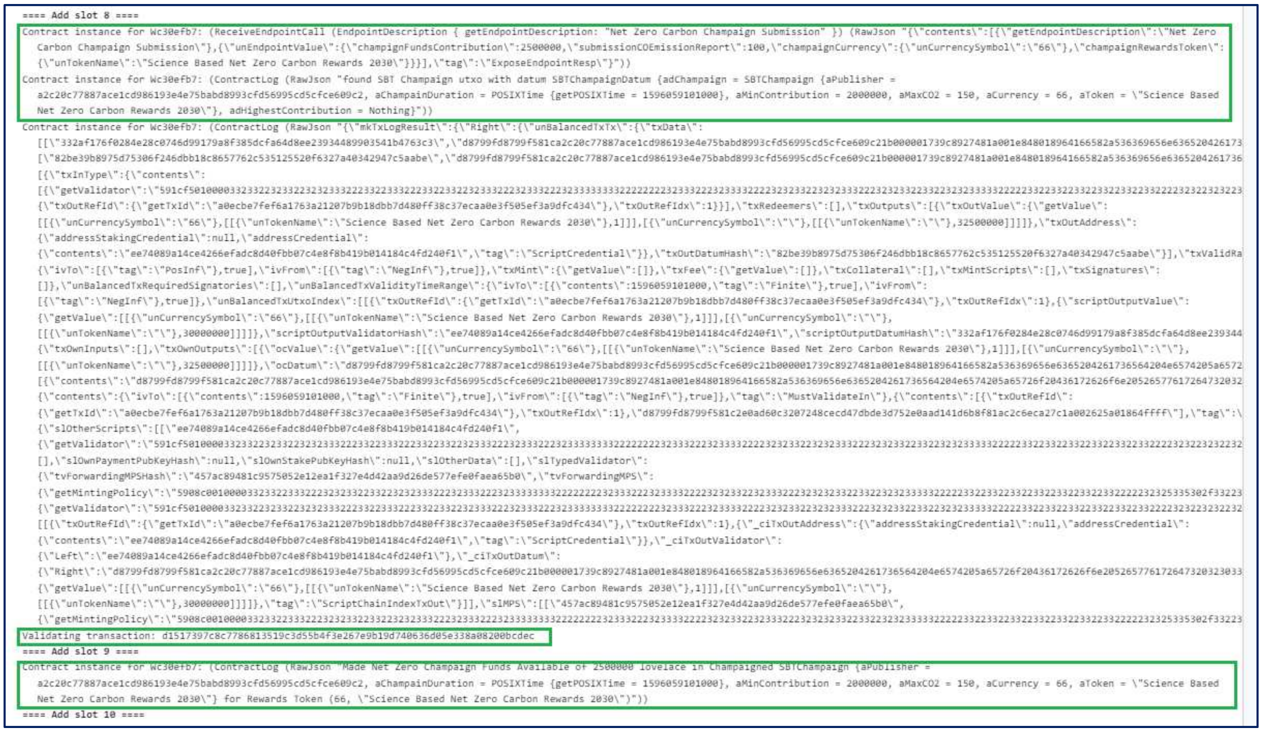

Figure 29 Smart Contract Simulation Log Info for Wallet 3 on Slot 8

4. Net Zero Carbon Campaign Smart Contract – Business Case and Development

4.1. Business Plan Executive Summary

Figure 30 Business Plan Executive Summary

Current features Carbon Decentralized Exchanges and Net Zero Carbon Campaign has been initiated as part of centralized campaign which have characteristic such as generalized in campaign, limited in participation, not open and transparent in campaign process, problem in scalability and interoperability, token offered based on trading functionality.

As part the observation on current net zero carbon campaign, we have listed identified problem and challenges which will further developed Business Plan Executive Summary as guidance for development journey of Net Zero Campaign Decentralized Application with better features.

The problem and challenge identification through evaluation mapping table including Value Journey Matrix, SAVE BEST Values, and MEET Evaluation to get clear pictures related the applicable facing problem and challenges which in-line to final product outcome whereas the identified tabulation defined and listed Solution Expectation and Experiences based on matrix (Product Relevance, Distinctiveness, and Salience). More-over Product Distinctiveness Matrix will give information to identify and gathered product available/current information to get more pictures the benefit and gap as part of Competitor Analysis (Carbondex, 2022).

Further evaluation is related to customer segment of the product through PIED Evaluation Table and Stakeholders Product Output matrix which defined and identify information of Proponent, Involvement, End User, and Decision Maker prior to develop and launching the product. This evaluation would be beneficial to map early adoption and future stakeholders which will be part of the ecosystem.

As part of Market Analysis, the evaluation going through customer observation as baseline for developing digital marketing strategy through Plan, Reach, Act, Convert, and Engage. This evaluation captured under Product Value Journey Matrix.

Further all the gathered information as part of business plan evaluation altered as baseline information to develop Business Ecosystem Lean Canvas.

As part of maintaining and monitoring the progress of development from start to full live product launching, with references to the information gathered through business evaluation, we develop Feasibility Road Map as part of Key Performance and Development Progress Monitoring.

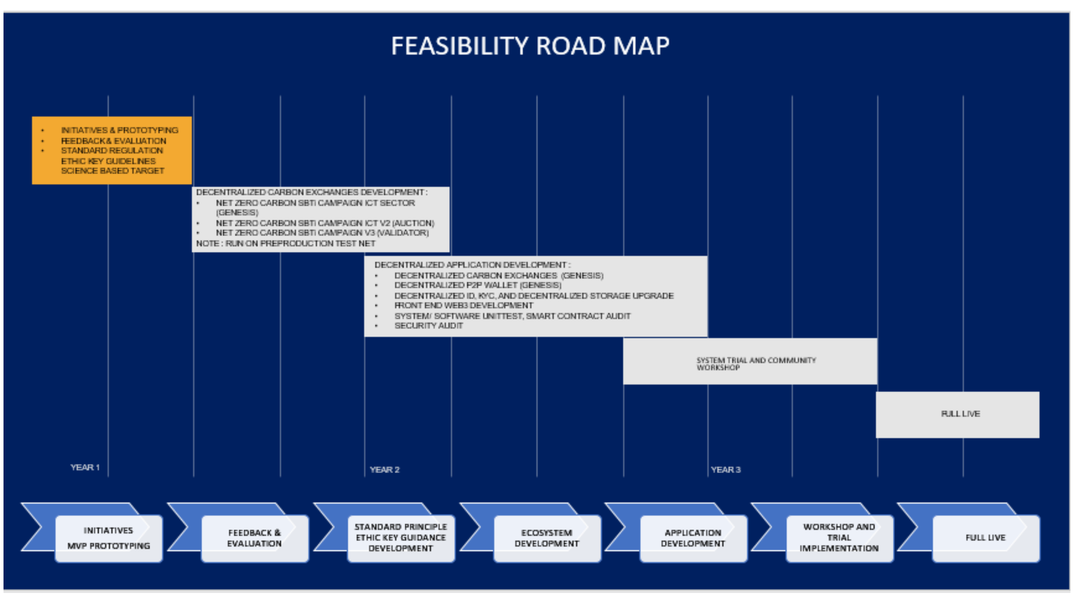

Figure 31 Feasibility Road Map

Carbon Trading Pools (DEX)
Net Zero Carbon SBTi Reward Campaign for ICT Company
BUSINESS ECOSYSTEM LEAN CANVAS

Problem:	Solution	Unique Value Propositions	Unfair Advantage	Customer Segments
i. Specific Net Zero Carbon Campaign Initiatives which are not open transparent auditable and trusted. ii. Scalability and Interoperability iii. Not transparent in Rewards (Fund and unique Token) iv. Not open in community participant v. Membership based on requirement and personnel interest. ***Existing Alternate Solution:*** *i.* Campaign through Centralized platform *ii.* Token based to trading functionality.	i. Campaign based on Specific Sector & Science Based Approached ii. Open Participant based on key interest. ***Key Metrics*** i. Developer Team ii. Business Team iii. Community iv. Registered Stake Member v. Registered Verified Auditor	Features: • Specific Sector Smart Contract Model/ Algorithm based on Science Based Approached and Community Verification • Multistage Campaign log info for each participant which made platform more open and transparent in deliver the campaign. • Simple safe secured ecosystem and easy to upgrade the smart contract (scalable) • Web3 and Digital Twin Technology Adoption • Able to integrate with third party application as part of calculation contributor. • Available rewards in Fund incentive and unique NFT • Scale Up ready to industry 5.0 Architectures (Secure and Safe Collaborative Artificial Intelligence Models) ***High Level Concepts*** • *Carbon Exchange – Net Zero Emission Campaign Metaverse* • *Carbon Exchange – Net Zero Emission Campaign Digital Twin*	i. Generative AI emission Validator ii. Complex Smart Contract for Other Specific Sectors ***Channels*** i. Web3 Internet Browser ii. Mobile App	i. B2B ICT Sectors ii. B2C ICT Sectors iii. IT and Digital Technology Providers iv. Verified Auditor & Regulator v. Net Zero Carbon Community ***Early Adopters*** *i.* B2B ICT Sectors *ii.* B2C ICT Sectors *iii.* Net Zero Carbon Community

Cost Structure	Revenue Streams
i. Decentralized Platform Leased ii. Insurance iii. Fund and Token Reward iv. Staff and facilities v. Technology Upgrade vi. Operation and Regular Audit vii. Taxes	i. Campaign Participant Registration ii. Stake Profit Funds iii. Campaign Winner Contribution Fee iv. Zero Emission Ads on Web3 platform v. Annual Global Emission Audit Released Report Funds

Table 5 Business Case Lean Canvas

4.2. Net Zero Carbon DApps Business Case – Problem Highlight

The ICT industry has been recognized as a noteworthy contributor to worldwide greenhouse gas (GHG) emissions. The proliferation of technology and the growing dependence on digital services have resulted in a notable escalation in energy consumption and the concomitant release of emissions. The escalating apprehensions regarding climate change and global warming have necessitated businesses to undertake measures to curtail their carbon footprint and endeavor to achieve net-zero emissions objectives.

The ICT industry encounters a significant obstacle in the form of an absence of a transparent, efficient, and dependable system to monitor, authenticate, and encourage initiatives aimed at reducing emissions in accordance with the most recent climate science. The conventional methods employed for quantifying and disclosing emissions can be intricate, laborious, and susceptible to errors, thereby posing challenges for firms to evaluate their advancements and pinpoint scopes for enhancement.

ICT companies encounter internal obstacles in executing efficient emissions reduction tactics, in addition to the pragmatic difficulties of tracking and disclosing emissions. The advancement of sustainability can be impeded by factors such as organizational inertia, financial limitations, and insufficient knowledge regarding the advantages of sustainable practices. In addition, corporations may encounter difficulties in obtaining the requisite knowledge and proficiency for formulating and executing science-based objectives and tactics.

The lack of a comprehensive and evidence-based strategy for mitigating emissions adds further complexity to the procedure. The lack of a cohesive framework may lead organizations to implement divergent and unaligned tactics that prove ineffective in producing significant outcomes. The absence of a cohesive strategy may result in a deficient sense of responsibility and openness, posing a challenge for enterprises to exhibit their dedication to sustainability and establish credibility with their interested parties.

Carbon Trading Pools (DEX) ***Net Zero Carbon SBTi Reward Campaign for ICT Company*** ***Product "RELEVANCE"***	
What value are stakeholders seeking to achieve (Outcomes) - Rank in order of importance?	*What value does our solution deliver. Rank our capability of delivering.*
1. Carbon Trading Decentralized Exchanged Apps which can give more proactive participation, incentive, or interest to G2G - B2B – B2C as part of Global Net Zero Carbon Initiatives 2. Carbon Trading Decentralized Exchanged Apps which have openness, secured, transparent, auditable, and trusted net zero carbon DLT records. 3. Carbon Trading Decentralized Exchanged Apps which have or bridging the fair incentives between the participant where the achievement results is open transparent auditable and trusted. 4. Carbon Trading Decentralized Exchanged Apps which have easy access and assess meaningful quality information to progress decision. 5. Carbon Trading Decentralized Exchanged Apps which have robust and effective report abnormality early detection.	1. One of Net Zero Carbon Trading Pools DApps Features is SBTi Zero Carbon Rewards Campaign Bounty Auction Algorithm where give more participation to B2B and B2C especially ICT Sectors. 2. SBTi Net Zero Carbon Smart Contract developed in On-Chain and Off-Chain. The insight reporting will be as part of Off-Chain DLT Metadata which are more secured and transparent. 3. SBTi Net Zero Carbon Smart Contract give the incentives as part of the Rewards Achievement in Rewards Fund and Unique NFT Rewards which further will adopt dynamic NFT. 4. SBTi Net Zero Carbon DApps metadata insight running on Off-Chain with Json format which further could be integrated in Web3. 5. Net Zero Carbon Trading Pools DApps running in most safe and secured 3rd generation blockchain ecosystem and metadata save in decentralized storage.
Where are the matches between problems/opportunities and solutions. *Are our strengths matched with their needs?* *Where are the gaps, and how important are they?*	
1. As Overall Stakeholder outcomes and Solution Deliver has matched on several items. 2. The gaps related to G2G participation which still not cover under preliminary stage of development. 3. Stakeholders System Architecture required adjustment to meet the DApps requirements. 4. Required to review the compliance to standard specification and law regulation documents related the technology solution delivered as part to maintain open transparency and trust. 5. Cross function with current other similar DApps system potentially will create conflict of interest, which required routine regular audit of the Smart Contract, Report Validation, and Law Regulation as part of Global Net Zero Carbon Incentives.	

Table 6 Solution Expectations and Product Value Journey Experiences Matrix (Relevance)

The absence of a worldwide platform for collaboration and communication is a significant obstacle confronting the ICT industry. While numerous enterprises may have comparable objectives and obstacles, they also exhibit unique characteristics and circumstances. At present, there exists no centralized platform that facilitates the pooling of resources, exchange of ideas, and mutual learning among stakeholders with regards to emissions reduction. The phenomenon of fragmentation poses a challenge to the industry's ability to foster innovation and expedite the achievement of net-zero emissions objectives.

The present scenario is characterized by the absence of a cohesive framework that can facilitate the resolution of obstacles faced by ICT enterprises and promote a collective sense of accountability in addressing the issue of climate change. To establish a transparent, efficient, and reliable system for tracking, verifying, and incentivizing emissions reductions efforts, a collaborative and integrated approach is necessary. This approach should utilize state-of-the-art technology and the most recent climate science.

To tackle these challenges, it is imperative to devise a solution that not only enables precise quantification and disclosure of emissions data but also fosters cooperation and dissemination of information among ICT enterprises. Through the utilization of technological capabilities and the implementation of contemporary developments in climate science, it is feasible to establish a more integrated and synchronized strategy for mitigating emissions. This approach would facilitate businesses in surmounting both internal and external obstacles, thereby propelling substantial advancement towards achieving net-zero emissions objectives.

An optimal resolution would furnish ICT enterprises with the necessary instruments and resources to devise and execute evidence-based objectives and tactics, while simultaneously presenting an open and verifiable documentation of their emissions abatement endeavors. The implementation of such measures could potentially enhance accountability and foster trust among stakeholders, while also promoting equitable opportunities for enterprises that prioritize sustainable practices.

Carbon Trading Pools (DEX) Net Zero Carbon SBTi Reward Campaign for ICT Company Product "DISTINCTIVENESS"	
The Main Competitors Known Value RANK order of importance	The Status Quo value & Common Practices Rank in order of demand
1. Maturity in Carbon Trading or Exchanges service with law and regulation compliance	1. Current legal and regulation which not based to Science Based Calculation will lead to dispute, lower trust and transparent.
2. Focussed on multi-level of specific sector and categories.	2. Generalization on Campaign will potential problem with the trust, transparency, and accountability of the report.
3. Campaign based on membership.	3. Campaign could be followed by any B2B or B2C without applying on membership, or anyone could participate as part of the net zero campaign.
4. Payment of participation through Debit/ Credit Card	4. Payment trough safe and secured decentralized platform based on wallet public beneficiary.
5. KYC through participant filling platform	5. KYC through safe and secured decentralized platform
Difference of Proposed Solutions Compared to Competitors (Status Quo, The Gaps, and Important)	
1. Net Zero Carbon Campaign Smart Contract worked based on Science Based Target Initiatives which involved trusted registered environment auditor and generative AI as campaign validator. As current Version 2 the validator through Science Based Target Calculation where all participants could overview the validation result. This will have made the campaign more open transparent auditable and trusted. 2. Net Zero Carbon Campaign Smart Contract focussed on each sector (currently ICT Sector) with several validation techniques where the process and result could be monitored by all participants. 3. Net Zero Carbon Campaign Smart Contract apply validation from public beneficiary wallet address which linkage to ICT company KYC information which lead to every company could participate in the campaign. 4. Net Zero Carbon Campaign used safe and secured P2P payment decentralized platform. 5. KYC utilized decentralized registration process and platform which stored in safe and secured decentralized storage.	

Table 7 Solution Expectations and Product Value Journey Experiences Matrix (Distinctiveness)

Carbon Trading Pools (DEX) Net Zero Carbon SBTi Reward Campaign for ICT Company PRODUCT SALIENCE	
What values so stakeholders seek in their outcome? Rank these in order of importance	How are we distinctively different from competition & status quo. Of our values what resonates? Rank in order of our Capability
1. Better technology solution	1. Technology solution offered more robust technology solution which develop in safe secured blockchain technology, specific use case and science-based technology validation methodology.
2. Easier to operate and analyse the insight.	2. Application develops in Web3 which could run on simple mobile application or digital Metaverse.
3. Safe secured open transparent and trusted ecosystem	3. Application develops in 3rd generation safe and secured blockchain ecosystem and metadata store in safe and secured decentralized storage. More-over status log could be monitor easily to make campaign more open and transparent.
4. Could deliver effective open transparent auditable and trusted insight.	4. Campaign result is open and transparent through Off-chain metadata.
5. Open Participation	5. Application adopts open community which defined as free user, registered campaign participant, registered data verifier/ validator, stake pool member, and registered auditor.
What is the SALIENCE fit between our Proposition and stakeholder desires? How well can we deliver what's needed? Where are the gaps, and how important are they?	
1. Currently Technology solution offered has more robust and cost optimization. 2. The gap related the resources, B2B, and G2G with stakeholder for custom model acquisition. 3. Stakeholder Standard Specification and Law Regulation related to technology solution should be adopted. 4. Integrating and aligned with current ecosystem. 5. Open Data sharing Regulation Compliance	

Table 8 Solution Expectations and Product Value Journey Experiences Matrix (Salience)

Competitor List:

- Centralized Campaign: Campaign Ad Net Zero Awards (Haymarket Media Group Limited, 2023)
- Decentralized Exchanges: Carbonex Exchange (Carbondex, 2022)

In addition, the cultivation of a collective obligation and cooperation through such a resolution may facilitate the dismantling of obstacles among enterprises and promote heightened ingenuity and advancement in the pursuit of ecological stability. Through collaborative efforts such as resource pooling, sharing of best practices, and knowledge transfer from both successful and unsuccessful experiences, ICT enterprises could expedite their endeavors to reduce emissions and generate a more substantial collective influence in the battle against climate change.

To summarize, the reduction of emissions in the ICT sector presents a complex and multifaceted set of challenges. To effectively address the obstacles hindering progress towards achieving net-zero emissions goals, a comprehensive and cooperative strategy is imperative. Through the utilization of contemporary technology and climate science, it is feasible to establish an integrated framework that facilitates the monitoring, validation, and incentivization of endeavors aimed at reducing emissions by businesses. Additionally, this framework cultivates a collective sense of accountability and cooperation among stakeholders. The implementation of such a system would enable enterprises to surmount both internal and external obstacles to the reduction of emissions, thereby fostering increased innovation and advancement in the global effort to combat climate change.

A plausible strategy for tackling these obstacles involves the creation of a blockchain-driven resolution that streamlines the process of monitoring, validating, and motivating endeavors aimed at reducing emissions. The utilization of blockchain technology presents various benefits in this scenario, such as enhanced transparency, immutability, and heightened security measures. A blockchain-based solution has the potential to enhance accountability and trust among stakeholders by establishing a decentralized and tamper-proof record of emissions data. This could facilitate businesses in demonstrating their dedication to sustainability.

Aware	***Trigger -> Aware -> Investigate.*** ***Net Zero Carbon Reward Campaign Pain Points:*** 1. *Non transparent Carbon Footprint and Actual Condition will increase potential bias on reporting and impact to environment.* 2. *Current methods, the campaign through centralized publisher which are not open and transparent how the campaign and selection of the winner is performed.* 3. *Community member could participate based on their choice.* 4. *Scalability and Interoperability to another platform* 5. *Safe Secured Open Transparent Auditable and Trusted Campaign*	Its Yes? then Progress
Believe	***Consider -> Set the Expectation -> Convince*** ***Digital Toolbox:*** 1. ***Calculator:*** *Net Zero Carbon Campaign Smart Contract developed through Science Based Target Initiatives Model which Calculated Carbon Emission Target through several specific calculated individual ICT Company Conditions.* 2. ***Diagnostic:*** *Net Zero Carbon Campaign Smart Contract developed in open and transparent log algorithm On-Chain and Off-Chain which could be analysed as part campaign and emission data reporting insight.* 3. ***Communicator:*** *Net Zero Carbon Campaign run on Web3 platform which could communicate the insight through various platform including cloud web base and mobile platform* 4. ***Recommender:*** *Net Zero Carbon Campaign Smart Contract will display recommended such as emission target and log information to participant and community.* 5. ***Advisor:*** *Net Zero Carbon Campaign back end and front end will show case advised control variable as part of prescriptive and optimization insight for global annual carbon emission performance based on Science Based Target Initiatives and Validated Emission Report.* 6. ***Benchmark:*** *Net Zero Carbon Campaign back end and front end can perform plant benchmark insight for global annual carbon emission performance based on Science Based Target Initiatives and Validated Emission Report.* 7. ***Simulator:*** *Since run on Web3 as part Digital Metaverse or Digital Twin, Net Zero Carbon Campaign back end and front end could perform simulation insight for global annual carbon emission performance based on Science Based Target Initiatives and Validated Emission Report which are more open transparent and trusted.*	Its Yes? then Progress
Commit	***Recommend -> Discuss -> Select -> Buy.*** ***Please see S.A.V.E.B.E.S.T Stakeholder Value Evaluation Table***	Its Yes? then Progress
Deliver	***Install -> Use -> Feedback -> Advocate.*** ***1. Please see M.E.ET Stakeholder Expectation Table*** ***2. Please see Stakeholder Prescriptive Outputs Evaluation Table***	Its Yes? then Progress

Table 9 Value Journey Matrix

A blockchain-based solution has the potential to facilitate the measurement and reporting of emissions data in a dependable manner. Furthermore, it could foster cooperation and knowledge exchange among ICT enterprises. The establishment of a collaborative platform among companies can facilitate the exchange of ideas, consolidation of resources, and assimilation of lessons from both successful and unsuccessful ventures. This approach can potentially foster innovation and advance the attainment of net-zero emissions objectives.

In addition, the integration of recent developments in climate science into a blockchain-based solution has the potential to assist businesses in devising and executing emissions reduction strategies that are more efficacious and grounded in scientific principles. This measure would facilitate the alignment of their operations with international climate objectives and guarantee the efficacy of their endeavors in combating climate change.

A potential approach to encourage businesses to engage in emissions reduction initiatives involves the integration of a blockchain-based solution that features a rewards system. This system would acknowledge and compensate companies for their advancements towards achieving net-zero emissions objectives. This measure has the potential to foster a more equitable and competitive business environment for enterprises that prioritize sustainable practices, while also serving as a catalyst for wider adoption of such practices.

To summarize, the implementation of a blockchain-oriented resolution that tackles the obstacles encountered by the Information and Communications Technology industry in relation to the mitigation of emissions possesses the capability to facilitate significant advancement towards the attainment of net-zero emissions objectives. Through the utilization of blockchain technology and contemporary developments in climate science, a prospective framework could potentially assist enterprises in surmounting both internal and external obstacles to the reduction of emissions, promoting cooperation and dissemination of knowledge, and establishing a more lucid and substantiated account of their endeavors.

Through the adoption of a more cohesive and cooperative methodology, companies operating in the field of Information and Communication Technology (ICT) have the potential to not only mitigate their own environmental impact but also make a meaningful contribution to the worldwide effort to combat climate change. By doing so, individuals or organizations can exhibit their dedication towards sustainability and establish themselves as pioneers in the shift towards a low-carbon economy. In the end, this endeavor will contribute towards establishing a sustainable future for the global community, wherein corporations assume a pivotal responsibility in addressing the most critical ecological issues of our era.

The ongoing development of a blockchain solution is intended to tackle various challenges associated with the attainment of Net Zero Carbon emissions.

The implementation of Net Zero Carbon Campaign Initiatives is characterized by openness, transparency, auditability, and trustworthiness. The blockchain solution aims to address a significant issue pertaining to the absence of open, transparent, auditable, and trustworthy Net Zero Carbon Campaign Initiatives. Numerous initiatives exhibit a deficiency in explicit and lucid objectives, approaches, and updates on advancement. This situation poses a challenge for stakeholders to furnish feedback and for initiatives to effectuate requisite modifications. The implementation of blockchain technology offers a framework for initiatives that are both transparent and auditable, thereby enabling stakeholders to oversee the advancement of such initiatives and furnish their input.

Scalability and interoperability are two important factors that need to be considered in the development of software systems. The blockchain technology offers a potential solution to the issues of scalability and interoperability that arise in the context of Net Zero Carbon Campaign Initiatives. Numerous initiatives face resource and infrastructure constraints that hinder their ability to generate substantial outcomes or engage in productive partnerships with other entities. The utilization of blockchain technology offers a viable platform for initiatives to expand their operations and collaborate with other entities in pursuit of shared objectives.

The lack of transparency in the rewards system, specifically regarding the fund and unique token, is a concern. The implementation of blockchain technology serves as a viable solution to the issue of insufficient transparency in the rewards systems employed in Net Zero Carbon Campaign Initiatives. The rewards systems of various initiatives often lack clarity, and the transparency of unique tokens is not always apparent. The implementation of a blockchain-based solution offers a transparent rewards system that incentivizes companies to achieve their Net Zero Carbon objectives.

The community participant is not granted access. The implementation of blockchain technology effectively tackles the issue of limited transparency in communal involvement. Numerous initiatives exhibit a deficiency in affording stakeholders with opportunities to provide their input and feedback, thereby impeding the effectiveness of collaborative efforts. The blockchain technology offers a platform that facilitates the participation of stakeholders in providing their input and feedback, thereby ensuring that initiatives align with the needs and interests of the communities they aim to serve.

Membership is determined by both eligibility criteria and individual interest. The blockchain solution effectively resolves the issue of membership in Net Zero Carbon Campaign Initiatives, which has traditionally been contingent upon fulfilling specific criteria rather than individual inclination. Several initiatives mandate skills or credentials yet fail to foster individual enthusiasm towards the mission and principles of the initiative. The implementation of blockchain technology offers prospects for individuals to engage in Net Zero Carbon Campaign Initiatives, contingent upon their eligibility and individual inclination towards the initiative's objectives and principles.

Several alternative solutions currently exist for addressing the challenges associated with achieving Net Zero Carbon emissions:

- The utilization of a centralized platform for campaigning. An alternative approach involves executing campaigns via centralized platforms. The process entails utilizing a centralized platform for the initiation and administration of Net Zero Carbon initiatives. This methodology could potentially enhance efficacy with regards to campaign oversight and monitoring. Nonetheless, it is possible that a centralized blockchain solution may lack the same degree of transparency and openness as a decentralized one, thereby potentially compromising the level of accountability and trust among stakeholders.
- The trading functionality is based on tokens. An additional viable solution entails the utilization of token-based systems that incorporate trading capabilities as a means of motivating Net Zero Carbon initiatives. The proposed methodology entails the generation of a distinct token that can serve as an incentive mechanism for corporations to incentivize their advancements in achieving Net Zero Carbon objectives. The tokens possess the capability

to be exchanged on digital currency platforms, thereby enabling enterprises to derive monetary value from their advancements towards attaining Net Zero Carbon emissions. Nonetheless, it is plausible that this methodology could lack the same level of transparency and efficacy as a blockchain-based resolution, given that the tokens may not be subject to the same degree of auditability or verifiability. Furthermore, this methodology may not foster cooperation and communal involvement to the same extent as a blockchain-oriented resolution, as it predominantly concentrates on the advancement of individual enterprises rather than the joint advancement towards achieving Net Zero Carbon emissions.

4.3. Net Zero Carbon DApps Business Case – Solution Proposal

The endeavor to tackle the obstacles in attaining Net Zero Carbon emissions is being pursued through the implementation of the subsequent methods:

The proposed solution entails implementing campaign-based strategies that are centered on specific sectors and grounded in scientific principles. The approach offers a precise and focused means of attaining the objective of Net Zero Carbon emissions. The solution can effectively tackle the unique challenges of each sector and promote progress towards Net Zero Carbon emissions by concentrating on specific sectors such as ICT and implementing science-based approaches.

The proposed solution incorporates an open participant-based strategy, whereby individuals are chosen based on their significant interest in the mission and values of the initiative. The aforementioned methodology guarantees the selection of stakeholders on the basis of their qualifications and personal investment in the initiative, thereby fostering increased levels of engagement and collaboration. This methodology additionally affords prospects for individuals hailing from varied backgrounds to partake in and make valuable contributions towards the triumph of the endeavor, thereby rendering it more comprehensive and representative.

4.4. Net Zero Carbon DApps Business Case – Value Creation

The value proposition of the blockchain solution is derived from its distinctive attributes and abstract principles.

Characteristics:

The proposed solution employs a distinct smart contract model/algorithm that is based on a scientific methodology and validated by the community. The practice guarantees that endeavors are focused and efficacious, with unambiguous and quantifiable objectives that are grounded on the most current scientific expertise.

The platform's transparency is enhanced by the provision of a multi-stage campaign log that furnishes information for every participant. The utilization of this log guarantees that interested parties can monitor the advancement of the project and furnishes a documentation of the initiatives undertaken during the campaign.

The solution's ecosystem is characterized by its simplicity, safety, and security, and it possesses the capacity to conveniently upgrade the smart contract to ensure scalability and adaptability to changing requirements. The solution also incorporates Web3 and Digital Twin technology, which enhances efficiency and transparency in monitoring and reporting emissions data.

The integration of the solution with third-party applications can offer supplementary tools and resources to support endeavors aimed at reducing emissions.

The proposed solution offers a system of incentives, including a fund incentive and unique NFTs, to encourage companies to make strides towards achieving their Net Zero Carbon objectives. Furthermore, the solution is equipped to be expanded to industry 5.0 architectures, utilizing secure and reliable collaborative artificial intelligence models to guarantee optimal data security.

The proposed solution involves the establishment of a Carbon Exchange - Net Zero Emission Campaign Metaverse using blockchain technology. The platform offers a digital environment for Net Zero Carbon endeavors to engage in cooperation, share expertise, and monitor advancements towards mutually agreed objectives.

The implementation also generates a digital replica of the Carbon Exchange - Net Zero Emission Campaign. The is a computerized depiction of an entity's greenhouse gas emissions profile, which facilitates enhanced efficacy and precision in the monitoring and reporting of emissions information.

4.5. Net Zero Carbon DApps Business Case – Customer Segment

The proposed solution can be advantageous for B2B ICT sectors as it involves the implementation of science-based strategies to reduce emissions and encourages enhanced cooperation and openness among stakeholders.

PIED Decision Framework	Attitude	Influence	Advocacy
Proponent			
1. Community Members	+/-	+/-	-
Involved:			
1. Free Community Member	+/-	+	+
2. Registered Campaign Participant	+	+	-
3. Registered Campaign Verifier/ Validator	+	+	+
4. Registered Stake Member	+	+	+
5. Registered Auditor	+	+	+
End User:			
1. B2B B2C ICT Sectors	+	+	-
2. Net Zero Carbon Community	+/-	+/-	-
3. Verified Emission Auditor	+	+	+
Decision Maker:			
1. Delegated Voted Stake Member	+	+	+
2. Verified Registered Active Community Member	+	+	-
3. Board of Management	+	+	+

Table 10 Proponent - Involved - End-user - Decision Maker Table

The solution can be advantageous for B2C ICT industries as it allows them to exhibit their dedication towards sustainability and diminish their carbon emissions in a clear and quantifiable manner.

The solution can be utilized by IT and digital technology providers to offer supplementary tools and resources for initiatives aimed at reducing emissions, as well as to demonstrate their dedication to sustainability.

The proposed solution can be advantageous for verified auditors and regulators as it can be utilized to authenticate emissions data and ensure adherence to Net Zero Carbon regulations.

The Net Zero Carbon community can derive advantages from the solution by utilizing it as a forum for cooperation, dissemination of knowledge, and monitoring advancements towards mutually agreed objectives.

The solution is expected to be adopted early by B2B ICT sectors due to their strong motivation to minimize their carbon footprint and their potential capacity to execute scientific methods for reducing emissions.

The B2C ICT industries could potentially serve as early adopters, given their increasing concern for sustainability and desire to showcase their dedication towards achieving Net Zero Carbon emissions.

The community that aims to achieve Net Zero Carbon is expected to be among the first to embrace the blockchain solution, given their existing commitment to reducing carbon emissions. They may perceive this technology to expedite the realization of their Net Zero Carbon objectives.

4.6. Net Zero Carbon DApps Business Case – Application Advantages

The blockchain solution possesses an inequitable advantage over alternative solutions for the following reasons:

1. The proposed solution integrates a generative artificial intelligence emission validator that can effectively authenticate emissions data in real-time. The validator offers superior precision and efficacy in contrast to conventional manual verification techniques, resulting in a distinct advantage in terms of both swiftness and precision.
2. The intricate smart contract of the solution enables the development of bespoke models tailored to industries, thereby facilitating a focused and efficacious strategy for mitigating emissions. The ability to customize the solution enables it to effectively address the distinct challenges encountered in various sectors, thereby conferring a competitive edge over other blockchain solutions that adopt a uniform approach.
3. The Smart Contract facilitates more participation to B2B - B2C ICT Sector and Community as part of campaign participant, verifier/ validator, auditor, and stake pools.
4. The Application develop in stage wise and publish on pre-production and production network to give more experiences to the community and received feedback for improvement prior full live stage where would be published on Main Net.
5. The Net Zero Carbon Campaign develops in each specific Sectors with script algorithm built On-Chain and Off-Chain where each transaction could be reviewed and audited by registered participant which led to open transparency and trust of the campaign.
6. The Campaign Rewards consist of rewards incentive fund and unique NFT which further will be upgraded as part of dynamic NFT rewards.
7. The Smart Contract develops in On-Chain and Off-Chain where all the log info as part transparency published in Off-Chain give more openness and transparency on each campaign transaction which gives more accountability and trust.
8. The Smart Contract Log Info develops multistage error log which makes it easy for the community to review and audit the transaction.
9. The Application will be full live launch in 3rd generation blockchain ecosystem which adopt dynamic P2P technology where enables automated connection of Stake Pool Operator relays to each other through self-discovery and optimization, removing the need of static configuration and manual maintenance required by the hybrid setup. This makes the network more performant and resilient to changes such as node or routing failures and streamlines the flow of information between the thousands of distributed nodes.

In general, the integration of generative artificial intelligence (AI) emission validation and adaptable smart contracts confers a distinct edge over alternative blockchain solutions, facilitating a more precise and focused strategy towards mitigating emissions.

	Better	Easier	Simpler	Tangible
Solution	Net Zero Carbon Campaign DApps Model Insight based on Science Based Target Initiatives and Validated Emission Report which are more open transparent and trusted.	Net Zero Carbon Campaign DApps Back End and Front-End Analysis Model easier to operated and understandable since based on based on Science Based Target Initiatives and Validated Emission Report and Web3 adoption	Community member or participant could independently evaluate the emission report submission since using open Science Based Target calculation which open to community.	Global Carbon Emission Annual live time insight could be visualized through current live data validated reports and based to the best performance log which more reliable and effective as part of Global Emission monitoring
Access	Net Zero Carbon Campaign accessibility is customized based on community member registration requirements type such as free member, registered campaign participant, registered data verifier/ validator, stake pool member, and registered auditor including secure access	Easier Access could be developed customized and modular based on requirements (backend and front end)	Net Zero Carbon Campaign Target Access Verification Model is simpler through KYC and Wallet Credential	Net Zero Carbon Campaign Access could be integrated and further developed as part ZK Proof Authentication Adoption.
Value Exchange	Net Zero Carbon Campaign with specific sector such as ICT will brough the Prescriptive Insight more open transparent auditable and trust as part global carbon emission monitoring.	Net Zero Carbon Campaign with specific sector and Science Based Approach will bring the Prescriptive Insight could be visualized easily through simple data visualization or Web3 Digital Metaverse Insight.	Net Zero Carbon Campaign with specific sector and Science Based Approach will bring the Apps Development could be develop as simpler model and could be visualized easily through simple data visualization or Web3 Digital.	Net Zero Carbon Campaign with specific sector and Science Based Approach Prescriptive Model could be initiate as Root Cause Analysis which could be used as part digital twin emission. Live Time Monitoring to detect any Green House abnormality based on Validated Emission Report.
Engagement	Net Zero Carbon Campaign will bring optimized community engagement including of Subject Matter Expert, B2B B2C ICT Sector, etc.	Net Zero Carbon Campaign have easy engagement since open to all community members.	Net Zero Carbon Campaign have simpler to attract community engagement based on their interest.	Net Zero Carbon Campaign brought the community engagement based on community interest and simpler KYC/ Wallet Credential Information

Table 11 S.A.V.E.B.E.S.T Values Evaluation Matrix

Motivators Value	Eliminate Bad Hygiene	Ease Value	Tangible Value
Better Technology Solution	Campaign delivered not Open and Transparent in data verification and validation	Technology easy to operate and adopt by community	Robust Model Apps (involved Community and Generative AI as part Node Validator)
Ease to operate, analyse, verified, and auditable	Complicated to operate and analyse	Campaign Report easy for community review and analyse	Platform easy to operate, analyse, verified, and auditable
Safe Secured Open Transparent Auditable and Trust Ecosystem	Campaign running on Ecosystem which are not Safe Secured Open Transparent Auditable and Trust	Campaign Application easy to operate in safe secured open transparent auditable and verifiable	Application running on top of Safe Secured Open Transparent Auditable and Trust Ecosystem
Robust, Effective and Efficient Insight	Complicated Insight	Campaign Report Insight or Log Report easy to interpret	Application develops with Robust, Effective and Efficient Insight where Campaign Report Insight or Log Report easy to interpret
Higher Performance in Scalability and Interoperability	Lower Performance in Scalability and Interoperability	Campaign Application easy to operate even though running by multiple users and interoperated with another platform	Application develops in Platform with high Performance in Scalability and Interoperability

Table 12 M.E.E.T Evaluation Table

		Net Zero Carbon Campaign for ICT Company Outputs					
		Purchase	Delivery	Use	Supplements	Maintenance	Disposal
Stakeholder Outcomes	Reduced anxiety	+	+	+	+	+	+
	Personal Benefits/ Incentive	+	+	+	+	+	+
	Better Strategy/ Function	+	+	+	+	+	+
	Easy Access	+	+	+	+	+	+
	Closer Relationship	+	+	+	+	+	+
	Scalability & Interoperability	+	+	+	+	+	+
	Speed of Transaction	+	+	+	+	+	+
	Safe Secured Open Transparent Trusted	+	+	+	+	+	+
	Operations Features	+	+	+	+	+	+

Table 13 Stakeholders Prescriptive Outputs Table

Stakeholder Critical Outcomes

\+ Experiences Better Than Expectation
O Experiences Meet Expectation
x Experiences Worse Than Expectation

4.7. Net Zero Carbon DApps Business Case – Channels

The blockchain solution's worth is being disseminated to customer segments via two primary channels.

The blockchain solution will be made available through a Web3 internet browser, which will enable users to access the platform from any internet-connected device. The provision of accessibility guarantees that the solution is accessible to a diverse range of users, thereby enabling increased adaptability with regards to the modes and locations of platform access.

The blockchain solution will be made available through a mobile application, enabling users to conveniently access the platform while on the move. The proposed mobile application aims to offer a simplified and intuitive interface, facilitating user engagement in activities aimed at reducing carbon emissions and monitoring advancements towards achieving Net Zero Carbon objectives.

The utilization of a Web3 internet browser and a mobile application offers various avenues for the dissemination of the blockchain solution's value to different customer segments. These communication channels facilitate accessibility, flexibility, and user-friendliness of the solution, thereby fostering increased stakeholder engagement and participation.

4.8. Net Zero Carbon DApps Business Case – Key Metrics

The efficacy of the blockchain solution shall be evaluated based on the subsequent pivotal parameters:

The performance of the developer team will be evaluated based on the punctual completion of milestones and the caliber of the code generated. Additional metrics encompass the quantity of reported and resolved bugs or issues, the efficacy of the development process, and the capacity to accommodate evolving requirements.

The efficacy of the business team shall be gauged by quantifiable metrics such as the quantity of partnerships forged, the expansion of the user base, and the capacity to generate revenue. Additional metrics encompass measures such as the degree of contentment among customers, the level of involvement exhibited by users, and the frequency of customer retention.

The level of engagement and participation in the platform, the number of contributions made, and the level of collaboration among members will serve as metrics to evaluate the success of the community. Additional metrics comprise of user feedback and levels of satisfaction.

Software Development Life Cycle Adoption

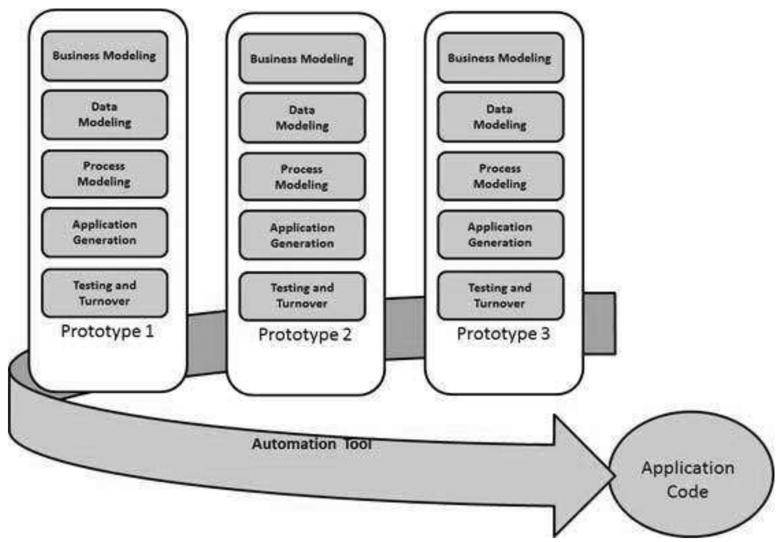

Figure 32 Rapid Application Development

The platform's efficacy will be evaluated based on the quantity of registered stake members who are actively engaged in initiatives aimed at reducing emissions, as well as their advancement towards achieving Net Zero Carbon objectives. This metric will determine the success of the platform. Additional metrics encompass the quantity of emissions reductions that have been verified and the magnitude of carbon offset.

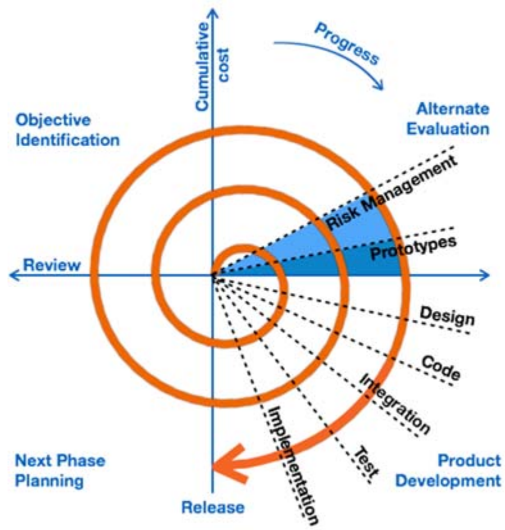

Figure 33 Spiral Model

The efficacy of the platform shall be evaluated based on the number of registered auditors who have been verified and are currently engaged in the verification of emissions data, as well as ensuring conformity with the Net Zero Carbon standards. These auditors shall be deemed registered and verified. Additional metrics encompass the precision and productivity of the authentication procedure and the degree of contentment among users.

The efficacy of the blockchain solution will be evaluated based on its capacity to proficiently tackle the obstacles associated with attaining Net Zero Carbon emissions, encourage cooperation and dissemination of knowledge among stakeholders, and furnish a lucid and streamlined framework for emissions reduction endeavors. The utilization of key metrics will facilitate the monitoring of progress and the identification of potential areas for enhancement, thereby ensuring the continuous evolution of the solution to cater to the requirements of its users.

4.9. Net Zero Carbon DApps Business Case – Cost Structures

The cost structure associated with the implementation of the blockchain solution will encompass the following components:

The foremost expense linked with the execution of the solution pertains to the acquisition of a decentralized platform lease. The cost of the service is subject to variation based on the provider and the features that are deemed necessary.

The cost of insurance is a crucial factor for any blockchain solution. The variability of insurance cost is contingent upon the requisite level of coverage and the hazards affiliated with the platform.

The expenditure associated with funding rewards and tokens is a crucial aspect to consider. The expenses incurred will be contingent upon the reward framework and the quantum of funds assigned to the rewards.

The expenses associated with personnel and infrastructure are contingent upon the magnitude and intricacy of the undertaking. The expenses associated with staffing and facilities, such as salaries and office space, may be encompassed within this cost.

The expense associated with technology upgrades may be imperative to ensure the platform remains current and satisfies the evolving demands of the market. The expenses incurred may encompass enhancements to hardware, software, and additional infrastructure.

The expenses associated with the continuous operation and periodic auditing of the platform represent a crucial aspect that warrants consideration. The cost of the platform will be contingent upon the degree of activity and the audit prerequisites.

The cost structure of a platform may be influenced by taxes, contingent upon the jurisdiction and tax regulations in place.

The total expenses associated with the deployment of the blockchain solution are contingent upon various factors such as the requisite functionalities, the degree of engagement on the platform, and the governing policies. The cost structure presents a fundamental framework for comprehending the primary cost determinants linked to the execution of the proposed solution.

4.10. Net Zero Carbon DApps Business Case – Revenue Streams

The following revenue sources are available for the blockchain solution:

The platform may charge a fee to register users to take part in efforts to reduce greenhouse gas emissions. Depending on the campaign and the level of engagement, this price may change.

Stake Profit Funds - The platform can make money by taking a cut of the profits made through stake funds, which are used to reward actions to reduce emissions. Depending on the campaign and the level of involvement, this proportion may change.

Campaign Winner Contribution Fee: If a campaign winner receives incentives or acknowledgment for their efforts to reduce emissions, the platform may charge a contribution fee. Depending on the campaign and the level of engagement, this price may change.

The Web3 platform can make money by selling advertising space to companies or groups that are dedicated to producing zero emissions. The Web3 platform allows for the presentation of this advertisement.

The platform can make money by publishing a yearly report on the global emission audit and charging a fee for access to the report. These revenue streams offer a range of ways to profit from blockchain technology while also encouraging initiatives to reduce emissions and advance a more sustainable future.

4.11. Net Zero Carbon DApps Features - KYC and AML

KYC is the process of verifying the identity of customers, while AML refers to a set of procedures, laws, and regulations designed to prevent money laundering activities. Since the main outcome of this project is develop Decentralized Carbon Exchanges with ability to perform Net Zero Carbon Campaign within specific Business Sector, the decentralized nature of platforms makes it difficult for regulatory bodies to monitor transactions and enforce compliance. Therefore, incorporating KYC logic algorithms into smart contracts can significantly enhance AML compliance efforts.

The benefits of implement and integrating KYC Logic Algorithms:

A. Enhanced Due Diligence: Smart contracts integrated with KYC algorithms (Cardano Feed, 2022) can streamline the collection and verification of user information, ensuring only verified users can participate in P2P Decentralized Stake Pools Carbon Exchanges (Cardano, 2021). This can significantly reduce the risk of illicit funds flowing through these platforms.
B. Improved Regulatory Compliance: By incorporating KYC algorithms, P2P Decentralized Stake Pools Carbon Exchanges platforms can demonstrate their commitment to adhere to AML regulations, potentially attracting more users and investors who seek a compliant ecosystem.
C. Increased Trust and Transparency: Implementing KYC logic in smart contracts fosters trust and transparency among platform users, as they can be assured that their fellow participants have undergone a thorough identity verification process.
D. Secured Liabilities: With KYC algorithms in place, lending pools can better manage their risks and protect their carbon exchanges assets from potentially fraudulent activities and fund defaults.

5. Net Zero Carbon Campaign Smart Contract – Plutus Backend and Hydra Environment

Science Base Target Net Zero Carbon Campaign Smart Contract developed in Plutus Language which deployed in test environment under Plutus Application Backend (Cardano, 2023) as part Cardano Blockchain Ecosystem (Greene, 2022).

Plutus Application Backend is the client-side runtime for Plutus apps that are built with the Plutus Platform (Cardano, 2023) to deal with requests from running Contract instances, to forward user input to them, and to notify them of ledger state change events.

As part of Client Interface, Plutus Application Backend provides an HTTP and WebSocket interface for interacting with Contract instances where all operations, including starting new instances, calling endpoints on instances, and querying instance state, are performed using this API. Application developers can build their own frontends and server processes that make HTTP calls to Plutus Application Backend.

There are two deployment models available through hosted and in-browser. In the “Hosted PAB” scenario, the dApp provider / developer hosts an instance of the PAB alongside the chain index and an Cardano-Alonzo node. The off-chain code of the Plutus app is run on the dApp provider’s infrastructure. In the “In-browser PAB” scenario, the dApp provider / developer hosts an instance of the chain index and an Cardano-Alonzo node. The dApp users work with a browser interface which uses a light version of the PAB.

Further stage in application development by launched the application in Cardano Main-Net (Cardano, 2022) which comply to Hydra Protocols (Hydra: Head Protocol, 2022). Hydra is a family of protocols intended to provide the layer 2 scalability solution for Cardano. The Hydra Head protocol is the first protocol in this family — it is a suite of smart contracts and software that allows any group of participants to establish isomorphic, multi-party state channels (Hydra heads) with each other.

Hydra heads allow participants to transact with each other, using the funds that they brought into the state channels, without having to submit these transactions to Cardano’s main network (layer 1).

The final result of these interactions between Hydra head participants can be brought back to Cardano layer 1 by closing the Hydra head, which releases the funds inside the head to be used in other Hydra heads or otherwise on Cardano layer 1.

Transactions within a head have the same format and properties as transactions on Cardano layer 1 – they are *isomorphic*. In principle, this allows Cardano DApps to re-use a significant portion of their codebase when transitioning some of their smart contracts to use Hydra (Hydra: Head Protocol, 2022). One exception to this isomorphism property is that token minting/burning within a Hydra head cannot affect Cardano layer 1.

Hydra head can be achieved on Cardano layer 1 is through redistribution of existing tokens from the UTXOs committed to the Hydra head to new UTXOs (Kammerer, 2022). This is done according to the transactions that participants have consented to inside the head.

There are some differences between the consensus protocols of Cardano layer 1 and Hydra layer 2. This may require existing Cardano DApps to adapt their design so that they can maintain their desired behavior within the Hydra head environment. Unlike Cardano layer 1, the Hydra head participants must all remain online and responsive to each other, during the operation of the Hydra head, and all participants must acknowledge and agree to each transaction within the Hydra head for it to have an effect.

Applications can be run within a single head, or they can be spread out across a network of Hydra heads. When an application is spread across a network of Hydra heads, its global state can be evolved either through communication between Hydra heads (bypassing layer 1) or by synchronizing local head states to layer 1 and resolving them there.

It's identified the limitations of running general applications on the Cardano Layer-1 main network:

1. Transaction throughput is insufficient for high frequency/volume user interactions.
2. Transaction finality time is too slow.
3. Transaction costs are too high.

4. Data storage capacity is too low for data-rich applications.
5. It is hard to reliably chain transactions between several participants.

The first three limitations particularly affect the scalability and business viability of auction projects on Cardano:

- Low transaction throughput can limit the number of bids in auctions, which may prevent them from reaching their full sale price.
- Slow transaction finality can slow down the rhythm of auctions, reducing the excitement/enjoyment that bidders feel from participating in auctions.
- High transaction costs can cut into the profits of sellers and auction houses or increase participation costs for users.

With implementation of Hydra Protocols, there are hoped to realize the following benefits from using Hydra for auction or campaign projects:

1. Higher throughput and faster transaction finality would allow the bidding frequency to be increased in auctions.
2. Cheaper transaction fees (possibly zero fees) would reduce costs for bidders, sellers, and auction houses.
3. Isomorphic smart contracts would allow significant re-use of layer 1 smart contracts in layer 2 and potentially a flexible deployment between layer 1 and layer 2. This could reduce development and audit costs.

More-over there are identified the following limitations that are currently discouraging from pursuing a Hydra-based auction or campaign implementation on their own such:

1. It is unclear how to run zero-sum games within a Hydra head. In the simplified head protocol (currently implemented in the hydra-poc GitHub repository), any head participant at any time can veto the further evolution of the Hydra head. Exercising this veto power leaves other participants, no choice but to close the Hydra head to layer 1 with the last head state that all participants managed to agree on before the veto. For an auction run within a single Hydra head, this veto power is particularly problematic because it can prevent the auction

from having any acknowledged bids at all. Indeed, there is no intrinsic incentive for any bidder to sign any bids submitted by any other bidder within the Hydra head because the other bidder's bid increases the price that the bidder would have to pay to win the auction.

2. Only the participants listed in the Hydra head initialization can participate in a Hydra head (i.e., no new participants can join), and every head participant must remain active and responsive to other participants for the head to continue operating (ie, no participant can leave without closing the Hydra head for all participants). This is a significant limitation for single-head auctions because it is highly desirable to let new bidders join ongoing auctions. Furthermore, bidders would prefer not to be stuck in auctions that they are no longer interested in, which may even lead them to exercise their veto power to close the Hydra head prematurely – an undesirable outcome for single-head auctions.
3. Hydra head participants may only use the funds that they committed to the Hydra head before it opened. This limitation exists in the current implementation of Hydra (hydra-poc repository), but it will be lifted when incremental commits/de-commits are implemented later in the Hydra roadmap. For now, this limitation effectively caps the maximum reachable price in any Hydra-based auction that requires bids to be fully backed by bidder deposits. Furthermore, it reveals each bidder's maximum possible bid, which can be exploited by other bidders that have more funds committed to the Hydra head.

The indicated 3 (three) of these limitations would have to be sufficiently mitigated for a Hydra-based auction to be viable as a product to be launched on Cardano, but that it is particularly important to overcome the first limitation to consider providing a Hydra-based auction on auction/ campaign-based platforms.

6. Summary and Conclusion

The Information and Communication Technology (ICT) sector is encountering a noteworthy hindrance in the shape of a deficiency of a clear, effective, and reliable mechanism for overseeing, verifying, and incentivizing endeavors targeted at mitigating emissions. The aforementioned factors comprise internal barriers, namely organizational resistance to change, financial constraints, and inadequate understanding of sustainable methodologies. Moreover, the absence of a comprehensive and evidence-based approach to reducing emissions introduces additional intricacy to the process. Furthermore, the absence of a centralized platform hinders the facilitation of resource pooling, idea exchange, and mutual learning among stakeholders in relation to reducing emissions. The mitigation of emissions within the Information and Communication Technology (ICT) industry poses a multifarious and intricate array of obstacles.

To adequately confront these hindrances, a comprehensive and collaborative approach is imperative. The proposed approach entails leveraging cutting-edge technology and climate science to establish a comprehensive system that streamlines the processes of monitoring, validating, and incentivizing initiatives geared towards mitigating emissions. Furthermore, this framework fosters a communal sense of responsibility and collaboration among individuals involved in the process. The proposal suggests utilizing blockchain technology to enhance the efficiency of monitoring, validating, and incentivizing efforts aimed at mitigating emissions. The utilization of blockchain technology holds promise in augmenting transparency, immutability, and security protocols.

The implementation of a blockchain-based solution can enhance the reliability of emissions data measurement and reporting, promote collaboration and information sharing among ICT companies, and incorporate advancements in climate science. Furthermore, it has the potential to enable the synchronization of activities with global climate targets and ensure the effectiveness of their efforts in mitigating climate change. By implementing a more integrated and collaborative approach, ICT companies can effectively reduce their environmental footprint and play a significant role in the global fight against climate change. The blockchain solution endeavors to tackle the deficiency of open, transparent, auditable, and reliable Net Zero Carbon Campaign Initiatives. The framework provides a means for transparent and auditable initiatives, allowing stakeholders to monitor progress and provide feedback.

Furthermore, it presents a prospective resolution to the challenges of scalability and interoperability that emerge within the framework of Net Zero Carbon Campaign Initiatives. The utilization of blockchain technology presents a feasible framework for endeavors to broaden their activities and cooperate with other entities in the pursuit of common goals. The system provides a clear rewards mechanism that motivates corporations to attain their Net Zero Carbon targets. Furthermore, it enables the engagement of interested parties in contributing their perspectives and evaluations, and admission is contingent upon meeting qualifying standards and personal inclination. There are alternative solutions available to tackle the obstacles related to attaining Net Zero Carbon emissions.

The strategies entail the employment of a centralized platform to facilitate campaigning, the adoption of token-based mechanisms to incentivize Net Zero Carbon endeavors, and the integration of blockchain-based solutions. The implementation of these methods is being pursued as a solution to address the problem. The proposed solution involves the implementation of campaign-based strategies that are focused on sectors and based on scientific principles. The initiative employs an open participant-based approach, whereby individuals are selected based on their alignment with the mission and values of the program. The value proposition of the blockchain solution is derived from its utilization of a unique smart contract model/algorithm that is grounded in a scientific methodology and has been vetted by the community.

The suggested approach entails the creation of a Carbon Exchange platform that operates within a Net Zero Emission Campaign Metaverse, leveraging blockchain technology. The platform provides a virtual space for Net Zero Carbon initiatives to collaborate, exchange knowledge, and track progress towards collectively established goals. The tool produces a digital representation of an organization's greenhouse gas emissions profile, thereby improving the accuracy and effectiveness of emissions monitoring and reporting. The blockchain technology provides value by creating a secure, transparent, and efficient platform for Net Zero Carbon initiatives. The proposed solution exhibits potential benefits for various sectors, including B2B and B2C ICT industries, IT and digital technology providers, verified auditors and regulators, as well as the Net Zero Carbon community.

The early adoption of environmentally friendly practices is anticipated among B2B ICT sectors, as they possess a strong drive to reduce their carbon footprint and have the potential to implement scientific approaches for emission reduction. The community striving to attain Net Zero Carbon is anticipated to be among the early adopters of the blockchain technology. The integration of a generative artificial intelligence emission validator and adaptable smart contracts confers an inequitable advantage upon the blockchain solution in comparison to alternative solutions. The dissemination of the product is being carried out through two primary channels, namely a Web3 internet browser and a mobile application, to reach out to the customer segments. The communication channels enhance the ease of access, adaptability, and ease of use of the solution, thereby promoting heightened involvement and engagement of stakeholders.

To evaluate the efficacy of the concept, the blockchain resolution will be rendered accessible via a Web3 internet browser as well as a mobile application. The effectiveness of the platform will be assessed through the analysis of several factors, including the number of stake members who have registered and are actively participating in initiatives aimed at reducing emissions, the number of verified emissions reductions achieved, the number of auditors who have registered on the platform, the accuracy and efficiency of the authentication process, and the level of user satisfaction. The execution of the proposed resolution will incur expenses. The expenses related to the deployment of a blockchain-based solution encompass the procurement of a decentralized platform lease, insurance, funding rewards and tokens, personnel and infrastructure, technology upgrades, uninterrupted operation, regular platform auditing, and tax obligations.

The variability of these expenditures is subject to the degree of operational engagement and audit requirements, in conjunction with the governing jurisdiction and tax statutes. The overall costs related to the implementation of a blockchain solution are dependent on a range of factors, including but not limited to functionality, user involvement, and regulatory frameworks. The blockchain solution's revenue streams comprise of user registration fees, profits from staked funds, fees from campaign winners' contributions, revenue from advertising space, and the publication of an annual report. The revenue streams provide diverse avenues for generating profits through the utilization of blockchain technology, while simultaneously promoting efforts towards mitigating emissions and propelling progress towards a more sustainable future.

7. References:

Anon., 2021. *The Corporate Net-Zero Standard.* [Online] Available at: https://sciencebasedtargets.org/net-zero [Accessed March 2023].

Carbondex, 2022. *The Next Generation Carbon Credit Exchange.* [Online] Available at: https://carbondex.exchange/ [Accessed April 2023].

Cardano Feed, 2022. *Blockpass Partners with EMURGO to Provide On-Chain KYC to Cardano Blockchain Ecosystem.* [Online] Available at: https://cardanofeed.com/blockpass-partners-with-emurgo-to-provide-on-chain-kyc-to-cardano-blockchain-ecosystem-69635.html [Accessed March 2023].

Cardano, 2021. *Peer-to-peer (P2P) networking.* [Online] Available at: https://docs.cardano.org/explore-cardano/cardano-network/p2p-networking [Accessed March 2023].

Cardano, 2022. *About the Cardano network.* [Online] Available at: https://docs.cardano.org/explore-cardano/cardano-network/about-the-cardano-network [Accessed April 2023].

Cardano, 2023. *What is the PAB?.* [Online] Available at: https://plutus-apps.readthedocs.io/en/latest/plutus/explanations/pab.html [Accessed March 2023].

Cardano, 2023. *What is the Plutus Platform?.* [Online] Available at: https://plutus.readthedocs.io/en/latest/explanations/platform.html [Accessed March 2023].

Greene, J., 2022. *Cardano For the MAsses: A Financial Operating System for people who don't have one Paperback.*. s.l.:https://cardanobook.com/.

Haymarket Media Group Limited, 2023. *Campaign Ad Net Zero Awards.* [Online] Available at: https://www.campaignadnetzeroawards.com/ [Accessed April 2023].

Hydra: Head Protocol, 2022. *Hydra Head Overview.* [Online] Available at: https://hydra.family/head-protocol/core-concepts/ [Accessed March 2023].

Hydra: Head Protocol, 2022. *Implementation of the Hydra Head protocol.* [Online] Available at: https://github.com/input-output-hk/hydra [Accessed March 2023].

IOHK, 2023. *Implementing auction projects using Hydra.* [Online] Available at: https://iohk.io/en/blog/posts/2023/01/20/implementing-auction-projects-using-hydra/ [Accessed March 2023].

Kammerer, T., 2022. *Classic UTxO model.* [Online] Available at: https://developers.cardano.org/docs/stake-pool-course/handbook/utxo-model/ [Accessed March 2023].

Science Based Targets, 2021. *Guidance for ICT Companies Setting Science Based Targets.* [Online] Available at: https://sciencebasedtargets.org/resources/legacy/2020/04/GSMA_IP_SBT-report_WEB-SINGLE.pdf
[Accessed March 2023].

Science Based Targets, 2021. *Guidance for ICT Companies Setting Science Based Targets.* [Online] Available at: https://sciencebasedtargets.org [Accessed March 2023].

Science Based Targets, 2021. *SBTI Corporate Net-Zero Standard Criteria Version 1.0,.* [Online] Available at: https://sciencebasedtargets.org/resources/files/Net-Zero-Criteria-Draft-for-Public-Consultation-v1-0.pdf
[Accessed March 2023].

Science Based Targets, 2021. *Target Validation Protocol for Near-term Targets Version 3.0.* [Online]
Available at: https://sciencebasedtargets.org
[Accessed March 2023].

ANNEX 1 – Net Zero Carbon Campaign Genesis Smart Contract Version 1.0

```haskell
{-# LANGUAGE DataKinds                  #-}
{-# LANGUAGE DeriveAnyClass             #-}
{-# LANGUAGE DeriveGeneric              #-}
{-# LANGUAGE DerivingStrategies         #-}
{-# LANGUAGE FlexibleContexts           #-}
{-# LANGUAGE GeneralizedNewtypeDeriving #-}
{-# LANGUAGE LambdaCase                 #-}
{-# LANGUAGE MultiParamTypeClasses      #-}
{-# LANGUAGE NoImplicitPrelude          #-}
{-# LANGUAGE OverloadedStrings          #-}
{-# LANGUAGE RecordWildCards            #-}
{-# LANGUAGE ScopedTypeVariables        #-}
{-# LANGUAGE TemplateHaskell            #-}
{-# LANGUAGE TypeApplications           #-}
{-# LANGUAGE TypeFamilies               #-}
{-# LANGUAGE TypeOperators              #-}

{-# OPTIONS_GHC -fno-warn-unused-imports #-}

-- Simple Application Carbon Credit Scientific Based Targets Rewards Smart Contract

module CarbonCreditRewardsFilter where

-- Import Dependency Data Environment
import           Control.Monad        (void)
import           Data.Aeson           (ToJSON, FromJSON)
import           Data.List.NonEmpty   (NonEmpty (..))
import           Data.Map (Map)
import           Data.Map                  as Map
import           Data.Maybe (catMaybes)
import qualified Data.ByteString.Char8     as C
import           Data.Text            (pack, Text)
import           GHC.Generics         (Generic)

-- Import Dependency Plutus Environment
import qualified PlutusTx                  as PlutusTx
import           PlutusTx.Prelude
import           Plutus.Contract
import PlutusTx.Prelude hiding (pure, (<$>))
import Prelude qualified as Haskell

-- Import Dependency Ledger Environment
import           Ledger                    (Address, Validator, ScriptContext, Value,
scriptAddress, getCardanoTxId, ChainIndexTxOut(..), Datum (Datum), dataHash, Datum (..
```

```
), DatumHash (..), PaymentPubKeyHash, TxInfo, scriptContextTxInfo, txSignedBy,unPaymen
tPubKeyHash)
import           Ledger                   hiding (singleton)
import           Ledger.Tx (ChainIndexTxOut (..))
import qualified Ledger.Constraints       as Constraints
import qualified Ledger.Typed.Scripts     as Scripts
import           Ledger.Value             as Value
import           Ledger.Ada               as Ada

-- Import Dependency Playground Environment
import           Playground.Contract
import qualified Prelude
import qualified Prelude             as P
import Prelude (String)
import           Text.Printf         (printf)


-- Functor Definition
--
newtype HashedString = HashedString BuiltinByteString deriving newtype (PlutusTx.ToDat
a, PlutusTx.FromData, PlutusTx.UnsafeFromData)
--PlutusTx.makeLift ''HashedString


-----------------------------------------------------------------
-- | Carbon Credit Science Based Target Validation On-Chain Code
-----------------------------------------------------------------
minLovelace :: Integer
minLovelace = 2000000


-- | Carbon Credit Science Based Target Datum On-Chain Code
data SBTDatum = SBTDatum
    { beneficiary       :: !PaymentPubKeyHash           -
- Public Hash Key Beneficiary Address
    , set_BaselineYear  :: !Integer                     -
- Baseline Year Science Based Target
    , set_TargetYear    :: !Integer                     -
- Target Year Science Based Target
    , set_SBTTonCO2E    :: !Integer                     -
- Science Based Target CO2 Emission in Tonnage
    , set_ERF           :: !Integer                     -
- Target Year Emission Reduction Factor
    , set_rewardAmount  :: !Integer                     -
- Carbon Emission Reward Amount
```

```
    , set_pin            :: !Integer                     -
- Pin Reward Withdrawl Authorization
    } deriving Show

PlutusTx.unstableMakeIsData ''SBTDatum
PlutusTx.makeLift ''SBTDatum


-- | Carbon Credit Rewards Redeemer On-Chain Code
data CCRewardRedeemer = CCRewardRedeemer
    { val_cc         :: !Integer            -
- Validated Value of Current Year Carbon Emission in Tonnage
    , val_erf        :: !Integer            -
- Validated Value 0f Current Year Emission Reduction Factor
    , val_pin        :: !Integer            -
- Validated Pin Reward Withdrawl Authorization
    } deriving Show

PlutusTx.unstableMakeIsData ''CCRewardRedeemer
PlutusTx.makeLift ''CCRewardRedeemer


-- | Carbon Credit Emission Validation On-Chain Code
-
- Transaction Validation to released Carbon Rewards through beneficiarry address match
ing, pin rewards authorization matching, and
-- Carbon Emission shall be below Tonnage of Target Year Carbon Emission
{-# INLINABLE validateCCEmission #-}
validateCCEmission :: SBTDatum -> CCRewardRedeemer -> ScriptContext -> Bool
validateCCEmission (SBTDatum phk _ _ ssbt serf _ dpnt) (CCRewardRedeemer valcc valerf
vpnt) ctx =
                                                                  traceIfFalse
"Emission Target Not Achieved - Fund Rewards On Hold" $   valcc*valerf < ssbt * serf
&&
                                                                 (traceIfFalse
"Wrong Withdrawl Pin!" $ dpnt == vpnt) &&
                                                                  traceIfFalse
"Beneficiary's Signature Not Matched" signedByBeneficiary
  where
    info :: TxInfo
    info = scriptContextTxInfo ctx

    signedByBeneficiary :: Bool
    signedByBeneficiary = txSignedBy info $ unPaymentPubKeyHash $ phk
```

```
-- | Datum and Redeemer parameter types
data CCRewards
instance Scripts.ValidatorTypes CCRewards where
    type instance DatumType CCRewards = SBTDatum
    type instance RedeemerType CCRewards = CCRewardRedeemer

-- | The script instance is the compiled validator (ready to go onto the chain)
ccRewardsInstance :: Scripts.TypedValidator CCRewards
ccRewardsInstance = Scripts.mkTypedValidator @CCRewards
  $$(PlutusTx.compile [|| validateCCEmission ||])
  $$(PlutusTx.compile [|| wrap ||])
    where
      wrap = Scripts.wrapValidator @SBTDatum @CCRewardRedeemer


-----------------------------------------------------------------
-- | Carbon Credit Science Based Target Validation Off-Chain Code
-----------------------------------------------------------------

-- | The Address of the Carbon Credit Science Based Target Rewards
ccRewardsAddress :: Address
ccRewardsAddress = Ledger.scriptAddress (Scripts.validatorScript ccRewardsInstance)

-- | Parameters for the "Carbon Credit Science Based Target Rewards" endpoint
data SBTParams = SBTParams
    { rewardBeneficiary  :: !PaymentPubKeyHash          -
- Public Hash Key Beneficiary Address
    , baselineYear       :: !Integer                    -
- Baseline Year Science Based Target
    , targetYear         :: !Integer                    -
- Target Year Science Based Target
    , target_SBTTonCO2E  :: !Integer                    -
- Science Based Target CO2 Emission in Tonnage
    , target_ERF         :: !Integer                    -
- Target Year Emission Reduction Factor
    , amount_Rewards     :: !Integer                    -
- Carbon Emission Reward Amount
    , pin                :: !Integer                    -
- Pin Reward Withdrawl Authorization
    }
    deriving (Generic, ToJSON, FromJSON, ToSchema)

--  | Parameters for the "Carbon Emission Validation" endpoint
data CCEmissionParams = CCEmissionParams
```

```
    { val_Current_Emission_TonCO2E           :: !Integer    -
- Validated Value of Current Year Carbon Emission in Tonnage
    , val_Current_Emission_Reduction_Factor  :: !Integer    -
- Validated Value 0f Current Year Emission Reduction Factor
    , pin_validation                         :: !Integer    -
- Validated Pin Reward Withdrawl Authorization
    }
    deriving stock (Prelude.Eq, Prelude.Show, Generic)
    deriving anyclass (FromJSON, ToJSON, ToSchema, ToArgument)

-
- | The schema of the contract, with one endpoint to publish the problem with a Carbon
 Credit Science Based Target Rewards
--   and to submit Carbon Emission Validation Value
type CCRewardsSchema =
            Endpoint "Science Base Target Emission ICT Data Center Rewards" SBTParams

        .\/ Endpoint "Current Emission ICT Data Center Submission" CCEmissionParams

-- | The "Carbon Credit Science Based Target Rewards" contract endpoint.
ccsbtrewards :: AsContractError e => SBTParams -> Contract () CCRewardsSchema e ()
ccsbtrewards (SBTParams bnf baseyear targetyear sbttarget erftarget rewardAmt pnt ) =
do
    let datDatum = SBTDatum
                { beneficiary       = bnf                 -
- SBT Datum Public Hash Key Beneficiary Address
                , set_BaselineYear  = baseyear            -
- SBT Datum Baseline Year Science Based Target
                , set_TargetYear    = targetyear          -
- SBT Datum Target Year Science Based Target
                , set_SBTTonCO2E    = sbttarget           -
- SBT Datum Science Based Target CO2 Emission in Tonnage
                , set_ERF           = erftarget           -
- SBT Datum Target Year Emission Reduction Factor
                , set_rewardAmount  = rewardAmt           -
- SBT Datum Carbon Emission Reward Amount in ADA
                , set_pin           = pnt                 -
- SBT Datum Pin Reward Withdrawl Authorization
                }

    let tx    = Constraints.mustPayToTheScript datDatum $ Ada.lovelaceValueOf rewardAmt
    ledgerTx <- submitTxConstraints ccRewardsInstance tx
    void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
    logInfo @String $ printf "Science Based Target Carbon Credit Emission Rewards with
 Baseline Year %d and Target Year %d"  baseyear targetyear
```

```
    logInfo @String $ printf "Science Based Target Carbon Credit Emission is %d TonCO2
e"  (sbttarget * erftarget)
    logInfo @String $ printf "Carbon Credit Bonus Available Funds of %d and Token Rewa
rds to be credited to Company with below Carbon Emission SBT Achievement on Target Yea
r" rewardAmt

-- | The "Carbon Emission Validation" contract endpoint.
carbonemissionupdate :: AsContractError e => CCEmissionParams -
> Contract () CCRewardsSchema e ()
carbonemissionupdate (CCEmissionParams valccemission targeterfvalue pin_validation) =
do
    onow   <- currentTime
    opkh   <- ownPaymentPubKeyHash
    -- filter all incorrect datum ccsbtrewards scripts
    unspentOutputs <- Map.filter hasCorrectDatum <$> utxosAt ccRewardsAddress
    let datRedeemer = CCRewardRedeemer
                { val_cc    = valccemission              -
- Carbon Emission Redeemer Validated Value of Current Year Carbon Emission in Tonnage
                , val_erf   = targeterfvalue             -
- Carbon Emission Redeemer Validated Value 0f Current Year Emission Reduction Factor
                , val_pin   = pin_validation             -
- Carbon Emission Redeemer Validated Pin Reward Withdrawl Authorization
                }

    let tx = collectFromScript unspentOutputs datRedeemer
    ledgerTx <- submitTxConstraintsSpending ccRewardsInstance unspentOutputs tx
    void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
    logInfo @String $ printf "Congratulation !! This Year Carbon Emission Achievement
is %d TonCO2e " (valccemission*targeterfvalue)
    logInfo @String $ printf "Carbon Credit Rewards will be creditted to Account Benef
iciary if Validated Carbon Emission Target Year below Science Based Target Value "
      where
        hasCorrectDatum :: ChainIndexTxOut -> Bool
        hasCorrectDatum (ScriptChainIndexTxOut _ _ (Right (Datum datum)) _)    =
          case PlutusTx.fromBuiltinData datum of
          Just d  -
> valccemission * targeterfvalue < (set_SBTTonCO2E d) * (set_ERF d) && pin_validation
== (set_pin d)
          Nothing -> False
        hasCorrectDatum _ = False

-- | Carbon Credit Science Base Target Rewards endpoints.
endpoints :: AsContractError e => Contract () CCRewardsSchema e ()
endpoints = awaitPromise (ccsbtrewards' `select` carbonemissionupdate') >> endpoints
  where
```

```
    ccsbtrewards' = endpoint @"Science Base Target Emission ICT Data Center Rewards" c
csbtrewards
    carbonemissionupdate' = endpoint @"Current Emission ICT Data Center Submission" ca
rbonemissionupdate

mkSchemaDefinitions ''CCRewardsSchema
```

ANNEX 2 – Net Zero Carbon Campaign Upgrade Smart Contract Version 2.0

```haskell
{-# LANGUAGE DataKinds                  #-}
{-# LANGUAGE DeriveAnyClass             #-}
{-# LANGUAGE DeriveGeneric              #-}
{-# LANGUAGE DerivingStrategies         #-}
{-# LANGUAGE FlexibleContexts           #-}
{-# LANGUAGE GeneralizedNewtypeDeriving #-}
{-# LANGUAGE LambdaCase                 #-}
{-# LANGUAGE MultiParamTypeClasses      #-}
{-# LANGUAGE NoImplicitPrelude          #-}
{-# LANGUAGE OverloadedStrings          #-}
{-# LANGUAGE RecordWildCards            #-}
{-# LANGUAGE ScopedTypeVariables        #-}
{-# LANGUAGE TemplateHaskell            #-}
{-# LANGUAGE TypeApplications           #-}
{-# LANGUAGE TypeFamilies               #-}
{-# LANGUAGE TypeOperators              #-}

--------------------------------------------------------------------------------------
-- |  Decentralized Application Net Zero Carbon Champaign Rewards Smart Contract   |--
-- |  Plutus Developer : Sonny Dhamayana Rochman                                   |--
-- |  Copyright        : @2023                                                     |-
-
--------------------------------------------------------------------------------------
-- Description :
-
- Decentralized Application Net Zero Carbon Champaign Rewards developed under Plutus S
mart Contract
-
- as part of Net Zero Carbon Global Champaign to achieve offset ambitious emissions re
ductions targets
-
- in line with the latest climate science and It is focused on accelerating companies
across the world
-- to halve emissions before 2030 and achieve net-zero emissions before 2050.
-
- The Decentralized Application work principle through regular update and validation C
arbon Emission Reports.
-
- The reports submit by each participant of the Net Zero Carbon Global Champaign and t
he reports
-- will be revalidated by Authorized Community Validator.
-- The best Achiever will get rewards as amount of Funds and Spesific Token Rewards.
-
- To prevent unwanted released rewards or malicious validation, Application protected
by
```

```
-
- Public Hash Key Authorization, Authorized Pin Rewards Withdrawl, and Decentralized V
oted Validation.
-
- Each Participant could resubmit their reports several times and get revalidation unt
il reached
-- the time of Champaign duration Ended.

module NetZeroCarbonChampaignRewards where

-- Import Dependency Data Environment
import           Control.Monad        hiding (fmap)
import           Data.Aeson           (ToJSON, FromJSON)
import           Data.List.NonEmpty   (NonEmpty (..))
import           Data.Map             as Map
import           Data.Text            (pack, Text)
import           GHC.Generics         (Generic)

-- Import Dependency Ledger Environment
import           Ledger               hiding (singleton)
import qualified Ledger.Constraints   as Constraints
import qualified Ledger.Typed.Scripts as Scripts
import           Ledger.Value         as Value
import           Ledger.Ada           as Ada

-- Import Dependency Playground Environment
import           Playground.Contract  (IO, ensureKnownCurrencies, printSchemas, stage,
 printJson)
import           Playground.TH        (mkKnownCurrencies, mkSchemaDefinitions)
import           Playground.Types     (KnownCurrency (..))

-- Import Dependency Plutus Environment
import           Plutus.Contract
import qualified PlutusTx
import           PlutusTx.Prelude     hiding (unless)
import qualified Prelude              as P
import           Schema               (ToSchema)
import           Text.Printf          (printf)

-- Set Minimum Campaign Rewards Funds in Lovelace
minLovelace :: Integer
minLovelace = 30000000

-----------------------------------------------------------------------------------------
---
```

```
-- | Decentralized Application Net Zero Carbon Champaign Rewards Validation On-
Chain Code
-------------------------------------------------------------------------------------
---

-- Net Zero Carbon Champaign Rewards Publisher Instance Data
data SBTChampaign = SBTChampaign
    { aPublisher            :: !PaymentPubKeyHash          -
- Science Based Target Zero Net Carbon Emission Champaign Publisher Beneficiary Addres
s
    , aChampainDuration     :: !POSIXTime                  -
- Net Carbon Emission Champaign Time Duration
    , aMinContribution      :: !Integer                    -
- Net Carbon Emission Champaign Participant Minimum Contribution in ADA
    , aMaxCO2               :: !Integer                    -
- Science Based Target Zero Net Carbon Emission Champaign Minimum Target Achievement
    , aCurrency             :: !CurrencySymbol             -- Currency Symbol
    , aToken                :: !TokenName                  -
- Science Based Target Zero Net Carbon Emission Champaign Token Rewards Name
    } deriving (P.Show, Generic, ToJSON, FromJSON, ToSchema)

instance Eq SBTChampaign where
    {-# INLINABLE (==) #-}
    a == b = (aPublisher            a == aPublisher         b) &&
             (aChampainDuration     a == aChampainDuration  b) &&
             (aMinContribution      a == aMinContribution   b) &&
             (aMaxCO2               a == aMaxCO2            b) &&
             (aCurrency             a == aCurrency          b) &&
             (aToken                a == aToken             b)

PlutusTx.unstableMakeIsData ''SBTChampaign
PlutusTx.makeLift ''SBTChampaign


-- Net Zero Carbon Champaign Rewards Participant Instance Data
data ChampaignSBT = ChampaignSBT
    { bParticipant          :: !PaymentPubKeyHash              -
- Science Based Target Zero Net Carbon Emission Champaign Participant Beneficiary Addr
ess
    , bContribution         :: !Integer                        -
- Science Based Target Zero Net Carbon Emission Champaign Participant Contribution Fun
ds
    , bCO2Emission          :: !Integer                        -
- Science Based Target Zero Net Carbon Emission Champaign Participant CO2 Emission Rep
orts
```

```
    } deriving P.Show

instance Eq ChampaignSBT where
    {-# INLINABLE (==) #-}
    b == c = (bParticipant          b == bParticipant  c) &&
             (bContribution         b == bContribution c) &&
             (bCO2Emission          b == bCO2Emission  c)

PlutusTx.unstableMakeIsData ''ChampaignSBT
PlutusTx.makeLift ''ChampaignSBT

-- Net Zero Carbon Champaign Rewards Champaign Action Data
data ChampaignAction = MkChampaign ChampaignSBT | Close
    deriving P.Show

PlutusTx.unstableMakeIsData ''ChampaignAction
PlutusTx.makeLift ''ChampaignAction


-- Wrapped Net Zero Carbon Champaign Rewards Datum Data
data SBTChampaignDatum = SBTChampaignDatum
    { adChampaign           :: !SBTChampaign                    -
- Wrapped SBT Champaign Publisher Data : aPublisher, aChampainDuration, aMinContributi
on, aMaxCO2, aCurrency, aToken
    , adHighestContribution :: !(Maybe ChampaignSBT)            -
- Wrapped SBT Champaign Participant Data : bParticipant, bContribution, bCO2Emission
    } deriving P.Show

PlutusTx.unstableMakeIsData ''SBTChampaignDatum
PlutusTx.makeLift ''SBTChampaignDatum


data Champaigned
instance Scripts.ValidatorTypes Champaigned where
    type instance RedeemerType Champaigned = ChampaignAction
    type instance DatumType Champaigned = SBTChampaignDatum

{-# INLINABLE minContribute #-}
minContribute :: SBTChampaignDatum -> Integer
minContribute SBTChampaignDatum{..} = case adHighestContribution of
    Nothing      ->  aMinContribution  adChampaign
    Just ChampaignSBT{..} ->   bContribution + 1

{-# INLINABLE maxCO2Emission #-}
maxCO2Emission :: SBTChampaignDatum -> Integer
```

```haskell
maxCO2Emission SBTChampaignDatum{..} = case adHighestContribution of
    Nothing      ->  aMaxCO2  adChampaign
    Just ChampaignSBT{..} ->   bCO2Emission


-- Onchain Validator Smart Contract
{-# INLINABLE mkChampaignValidator #-}
mkChampaignValidator :: SBTChampaignDatum -> ChampaignAction -> ScriptContext -> Bool
mkChampaignValidator ad redeemer ctx =
    traceIfFalse "wrong input value" correctInputValue &&
    case redeemer of
        MkChampaign b@ChampaignSBT{..} ->
            traceIfFalse "Contribution Funds too low"        (sufficientBid bContribut
ion)         &&
            traceIfFalse "wrong output datum" (correctBidOutputDatum b)    &&
            traceIfFalse "wrong output value" (correctBidOutputValue bContribution) &&
            traceIfFalse "wrong refund"       correctBidRefund             &&
            traceIfFalse "too late"           correctBidSlotRange
        Close           ->
            traceIfFalse "too early" correctCloseSlotRange &&
            case adHighestContribution ad of
                Nothing      ->
                    traceIfFalse "expected seller to get token" (getsValue (aPublisher
 auction) $ tokenValue <> Ada.lovelaceValueOf minLovelace)
                Just ChampaignSBT{..} ->
                    traceIfFalse "expected highest Participant to get token" (getsValu
e bParticipant  $ tokenValue <> Ada.lovelaceValueOf minLovelace) &&
                    traceIfFalse "expected seller to get highest Contributor" (getsVal
ue (aPublisher auction) $ Ada.lovelaceValueOf bContribution)

  where
    info :: TxInfo
    info = scriptContextTxInfo ctx

    input :: TxInInfo
    input =
      let
        isScriptInput i = case (txOutDatumHash . txInInfoResolved) i of
            Nothing -> False
            Just _   -> True
        xs = [i | i <- txInfoInputs info, isScriptInput i]
      in
        case xs of
            [i] -> i
            _   -> traceError "expected exactly one script input"
```

```
    inVal :: Value
    inVal = txOutValue . txInInfoResolved $ input

    auction :: SBTChampaign
    auction = adChampaign ad

    tokenValue :: Value
    tokenValue = Value.singleton (aCurrency auction) (aToken auction) 1

    correctInputValue :: Bool
    correctInputValue = inVal == case adHighestContribution ad of
        Nothing      -> tokenValue <> Ada.lovelaceValueOf minLovelace
        Just ChampaignSBT{..} -
> tokenValue <> Ada.lovelaceValueOf (minLovelace + bContribution)

    sufficientBid :: Integer -> Bool
    sufficientBid amount = amount >= minContribute ad

    ownOutput   :: TxOut
    outputDatum :: SBTChampaignDatum
    (ownOutput, outputDatum) = case getContinuingOutputs ctx of
        [o] -> case txOutDatumHash o of
            Nothing   -> traceError "wrong output type"
            Just h -> case findDatum h info of
                Nothing        -> traceError "datum not found"
                Just (Datum d) ->  case PlutusTx.fromBuiltinData d of
                    Just ad' -> (o, ad')
                    Nothing  -> traceError "error decoding data"
        _   -> traceError "expected exactly one continuing output"

    correctBidOutputDatum :: ChampaignSBT -> Bool
    correctBidOutputDatum b = (adChampaign outputDatum == auction)   &&
                              (adHighestContribution outputDatum == Just b)

    correctBidOutputValue :: Integer -> Bool
    correctBidOutputValue amount =
        txOutValue ownOutput == tokenValue <> Ada.lovelaceValueOf (minLovelace + amoun
t)

    correctBidRefund :: Bool
    correctBidRefund = case adHighestContribution ad of
        Nothing      -> True
        Just ChampaignSBT{..} ->
          let
```

```
            os = [ o
                 | o <- txInfoOutputs info
                 , txOutAddress o == pubKeyHashAddress bParticipant  Nothing
                 ]
          in
            case os of
                [o] -> txOutValue o == Ada.lovelaceValueOf bContribution
                _   -> traceError "expected exactly one refund output"

    correctBidSlotRange :: Bool
    correctBidSlotRange = to (aChampainDuration auction) `contains` txInfoValidRange i
nfo

    correctCloseSlotRange :: Bool
    correctCloseSlotRange = from (aChampainDuration auction) `contains` txInfoValidRan
ge info

    getsValue :: PaymentPubKeyHash -> Value -> Bool
    getsValue h v =
      let
        [o] = [ o'
              | o' <- txInfoOutputs info
              , txOutValue o' == v
              ]
      in
        txOutAddress o == pubKeyHashAddress h Nothing

typedAuctionValidator :: Scripts.TypedValidator Champaigned
typedAuctionValidator = Scripts.mkTypedValidator @Champaigned
    $$(PlutusTx.compile [|| mkChampaignValidator ||])
    $$(PlutusTx.compile [|| wrap ||])
  where
    wrap = Scripts.wrapValidator @SBTChampaignDatum @ChampaignAction

sbtchampaignValidator :: Validator
sbtchampaignValidator = Scripts.validatorScript typedAuctionValidator

auctionHash :: Ledger.ValidatorHash
auctionHash = Scripts.validatorHash typedAuctionValidator

auctionAddress :: Ledger.Address
auctionAddress = scriptHashAddress auctionHash


-- Off Chain Input Data Script Parameters
```

```
data StartParams = StartParams
    { setChampaignDuration          :: !POSIXTime              -
- Science Based Target Zero Net Carbon Emission Champaign Duration
    , setMinFundContribution        :: !Integer                -
- Science Based Target Zero Net Carbon Emission Champaign Participant Minimum Funds Co
ntribution
    , setMaxCO2EmissionTarget       :: !Integer                -
- Science Based Target Zero Net Carbon Emission Champaign CO2 Emission Minimum Target
    , setCurrency                   :: !CurrencySymbol         -
- Science Based Target Zero Net Carbon Emission Champaign Currency
    , setRewardTokenName            :: !TokenName              -
- Science Based Target Zero Net Carbon Emission Champaign Rewards Token
    } deriving (Generic, ToJSON, FromJSON, ToSchema)

data BidParams = BidParams
    { champaignCurrency             :: !CurrencySymbol         -
- Science Based Target Zero Net Carbon Emission Champaign Currency
    , champaignRewardsToken         :: !TokenName              -
- Science Based Target Zero Net Carbon Emission Champaign Rewards Token
    , champignFundsContribution     :: !Integer                -
- Science Based Target Zero Net Carbon Emission Champaign Participant Funds Contributi
on
    , submissionCOEmissionReport    :: !Integer                -
- Science Based Target Zero Net Carbon Emission Champaign Validated CO2 Emission Repor
t Submission
    } deriving (Generic, ToJSON, FromJSON, ToSchema)

data ValidationParams = ValidationParams
    { valCurrency       :: !CurrencySymbol
    , valToken          :: !TokenName
    , valCO2Emission    :: !Integer
    } deriving (Generic, ToJSON, FromJSON, ToSchema)

data CloseParams = CloseParams
    { cpCurrency :: !CurrencySymbol
    , cpToken    :: !TokenName
    } deriving (Generic, ToJSON, FromJSON, ToSchema)

type SBTChampaignSchema =
        Endpoint "Net Zero Carbon Champaign Started"        StartParams
    .\/ Endpoint "Net Zero Carbon Champaign Submission"     BidParams
    .\/ Endpoint "Net Zero Carbon Champaign Ended"          CloseParams

-- Off Chain Campaign Started End Points
champaignstarted :: AsContractError e => StartParams -> Contract w s e ()
```

```
champaignstarted StartParams{..} = do
    pkh <- ownPaymentPubKeyHash
    let a = SBTChampaign
                { aPublisher   = pkh
                , aChampainDuration = setChampaignDuration
                , aMinContribution    = setMinFundContribution
                , aMaxCO2    =  setMaxCO2EmissionTarget
                , aCurrency = setCurrency
                , aToken     = setRewardTokenName
                }
        d = SBTChampaignDatum
                { adChampaign    = a
                , adHighestContribution = Nothing
                }
        v = Value.singleton setCurrency setRewardTokenName 1 <> Ada.lovelaceValueOf mi
nLovelace
        tx = Constraints.mustPayToTheScript d v
    ledgerTx <- submitTxConstraints typedAuctionValidator tx
    void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
    logInfo @P.String $ printf "started SBT Champaign %s for token %s" (P.show a) (P.s
how v)


-- Off Chain Campaign Bidding Submission End Points
champaign :: forall w s. BidParams -> Contract w s Text ()
champaign BidParams{..} = do
  (oref, o, d@SBTChampaignDatum{..}) <- findCampaign champaignCurrency champaignReward
sToken
  logInfo @P.String $ printf "found SBT Champaign utxo with datum %s" (P.show d)

-- Error Log and Unlocked Rewards Schema
  if (champignFundsContribution  < minContribute d)
    then logError @P.String $ printf "Contribution Funds lower than minimal Contributi
on Required as %d" (minContribute d)
    else if (submissionCOEmissionReport > maxCO2Emission d)
        then logError @P.String $ printf "CO2 Emission higher than maximum allowed in
the Campaign as %d tCO2/Year" (maxCO2Emission d)
    else do
        pkh <- ownPaymentPubKeyHash
        let b  = ChampaignSBT {bParticipant  = pkh, bContribution = champignFundsContr
ibution, bCO2Emission = submissionCOEmissionReport}
            d' = d {adHighestContribution = Just b}
            v  = Value.singleton champaignCurrency champaignRewardsToken 1 <> Ada.love
laceValueOf (minLovelace + champignFundsContribution )
            r  = Redeemer $ PlutusTx.toBuiltinData $ MkChampaign b
```

```
            lookups = Constraints.typedValidatorLookups typedAuctionValidator P.<>
                    Constraints.otherScript sbtchampaignValidator                P.<>
                    Constraints.unspentOutputs (Map.singleton oref o)
            tx      = case adHighestContribution of
                        Nothing      -
> Constraints.mustPayToTheScript d' v                              <>
                                        Constraints.mustValidateIn (to $ aChampainDura
tion adChampaign)          <>
                                        Constraints.mustSpendScriptOutput oref r
                        Just ChampaignSBT{..} -
> Constraints.mustPayToTheScript d' v                              <>
                                        Constraints.mustPayToPubKey bParticipant  (Ada
.lovelaceValueOf bContribution) <>
                                        Constraints.mustValidateIn (to $ aChampainDura
tion adChampaign)          <>
                                        Constraints.mustSpendScriptOutput oref r
        ledgerTx <- submitTxConstraintsWith lookups tx
        void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
        logInfo @P.String $ printf "Made Net Zero Champaign Funds Available of %d love
lace in Champaigned %s for Rewards Token (%s, %s)"
            champignFundsContribution
            (P.show adChampaign)
            (P.show champaignCurrency)
            (P.show champaignRewardsToken)


-- Champaind Ended Contract
champaignended :: forall w s. CloseParams -> Contract w s Text ()
champaignended CloseParams{..} = do
    (oref, o, d@SBTChampaignDatum{..}) <- findCampaign cpCurrency cpToken
    logInfo @P.String $ printf "Found Champaign utxo with datum %s" (P.show d)

    let t      = Value.singleton cpCurrency cpToken 1
        r      = Redeemer $ PlutusTx.toBuiltinData Close
        seller = aPublisher adChampaign

        lookups = Constraints.typedValidatorLookups typedAuctionValidator P.<>
                  Constraints.otherScript sbtchampaignValidator                P.<>
                  Constraints.unspentOutputs (Map.singleton oref o)
        tx      = case adHighestContribution of
                    Nothing      -
> Constraints.mustPayToPubKey seller (t <> Ada.lovelaceValueOf minLovelace)  <>
                                    Constraints.mustValidateIn (from $ aChampainDurati
on adChampaign)                <>
```

```
                                    Constraints.mustSpendScriptOutput oref r
                    Just ChampaignSBT{..} -
> Constraints.mustPayToPubKey bParticipant  (t <> Ada.lovelaceValueOf minLovelace) <>
                                    Constraints.mustPayToPubKey seller (Ada.lovelaceVa
lueOf bContribution)              <>
                                    Constraints.mustValidateIn (from $ aChampainDurati
on adChampaign)                    <>
                                    Constraints.mustSpendScriptOutput oref r
    ledgerTx <- submitTxConstraintsWith lookups tx
    void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
    logInfo @P.String $ printf "Closed SBT Champaign %s for token (%s, %s)"
        (P.show adChampaign)
        (P.show cpCurrency)
        (P.show cpToken)

findCampaign :: CurrencySymbol
            -> TokenName
            -> Contract w s Text (TxOutRef, ChainIndexTxOut, SBTChampaignDatum)
findCampaign cs tn = do
    utxos <- utxosAt $ scriptHashAddress auctionHash
    let xs = [ (oref, o)
             | (oref, o) <- Map.toList utxos
             , Value.valueOf (_ciTxOutValue o) cs tn == 1
             ]
    case xs of
        [(oref, o)] -> case _ciTxOutDatum o of
            Left _          -> throwError "Datum missing"
            Right (Datum e) -> case PlutusTx.fromBuiltinData e of
                Nothing -> throwError "Datum has wrong type"
                Just d@SBTChampaignDatum{..}
                    | aCurrency adChampaign == cs && aToken adChampaign == tn -
> return (oref, o, d)
                    | otherwise                                           -
> throwError "Champaigned token missmatch"
        _           -> throwError "Champaigned utxo not found"


-- Off Chain End Points Contract Schema
endpoints :: Contract () SBTChampaignSchema Text ()
endpoints = awaitPromise (champaignstarted' `select` champaign' `select` champaignende
d') >> endpoints
  where
    champaignstarted'       = endpoint @"Net Zero Carbon Champaign Started"     champa
ignstarted
```

```
    champaign'              = endpoint @"Net Zero Carbon Champaign Submission"  champa
ign
    champaignended'         = endpoint @"Net Zero Carbon Champaign Ended"       champa
ignended

mkSchemaDefinitions ''SBTChampaignSchema

myToken :: KnownCurrency
myToken = KnownCurrency (ValidatorHash "f") "Token" (TokenName "Net Zero Carbon Awards
" :| [])

mkKnownCurrencies ['myToken]
```