MIE354 - Assignment 2 - Coorest

Shreya Perumal	
Anjali Jammulamadaka	

Declaration of Generative AI Usage

“In submitting this Assignment 2, I confirm that the work I am submitting is the exclusive work of the students in my group, and that I made an equal contribution to the work of the group. I confirm that my conduct during this Assignment 2 adheres to the Code of Behaviour on Academic Matters. I confirm that I am abiding by the course policy regarding the use of Generative AI tools and that I did NOT act in such a way that would constitute cheating, misrepresentation, or unfairness, including but not limited to, using unauthorized aids and assistance, impersonating another person, and committing plagiarism. I pledge upon my honour that I have not violated the Faculty of Applied Science & Engineering’s Honour Code during this assessment.”

Course Policy

“Students are encouraged to make use of Generative artificial intelligence tools (e.g., Copilot, ChatGPT, or any other similar tool) to contribute to their understanding of course materials and to complete coursework submission in compliance with the general course instructions and any additional instructions provided in the coursework handouts. Students are ultimately accountable for the work they submit. The general instructions are that students must declare in their submissions the Generative AI tools that have been used and include a summary of how the tools have been used. Using Generative AI tools and failing to follow the required instructions can result in an "unauthorized assistance" academic offence.”

Release Statement

“I agree to make available my work developed through the course Assignments (not including my student name) for educational purposes.”

Files in Submission G04.zip

1) G04-ChDescription.docx - Report about Choreography
2) G04-ChModel.vsdx - Visio of Business Choreography

Table of Contents

1.0 Choreography Objectives, Scope, and Participants

The following section is broken into three sections: choreography objectives, scope, and participants, providing a brief introduction and background on the choreography described in the report.

1.1 Choreography Objectives

The choreography described in this report models the interactions between Coorest and its participants to issue, verify, and transfer carbon tokens. Coorest's primary objective is to improve accessibility and trust in carbon-token purchasing by automating and streamlining the purchase and trading process through NFTrees. Specifically, this choreography aims to:

- Automate carbon-token purchase and verification
- Accurately map NFTrees to real geographic carbon data
- Guarantee secure data transmission and storage
- Leverage smart contracts to facilitate token ownership and transferability
- Maintain carbon transparency

Coorest enhances transparency and credibility in the carbon market by replacing manual verification and record-keeping with an automated, blockchain-based process. The choreography ensures that carbon-offset purchases, NFTree creation, and registry updates occur through the secure and verifiable collaboration of many independent participants. The choreography models interactions among Coorest, PolygonPOS Blockchain, FileCoin Storage, Chainlink Oracle, Registry Blockchain Smart Contract, NFTree Customers, and Carbon Token Buyers, all of which support their objectives.

1.2 Choreography Scope

The scope defines the bounds of the choreographies outlined in the following report. The choreographies will focus on the interactions between Coorest and its main participants in completing the main processes required in the cycle of an NFTree. Below is the process of interest, defining the scope of the report.

- *NFTree Purchasing:* NFTree customers submit purchase requests for NFTrees that produce carbon tokens based on the carbon mitigation of real trees.
- *Carbon Data Validation:* PolygonPOS validates the carbon mitigation data supplied to Coorest by Chainlink Oracle and issues tokens.
- *NFTree Creation and Minting:* Planted trees are added to a registry where they are immutable and are minted with their own unique identities.
- *Data Storage:* FileCoin is used to store information like carbon data, token/tree ownership, NFTree data and token transferring.
- *Registry Recording:* Updating the registry NFTrees as they are planted and initialized.

- *Token Ownership and Transfer:* blockchain facilitates transparent ownership and transfer.

The scope of the report will exclude processes related to organizational logistics of participants, government policy & licensing, and other activities not directly related to the Coorest Registry.

1.3 Choreography Participants

The choreography for the Coorest system involves seven participants, each representing an entity that contributes to the processes (outlined in the scope) of the Coorest system. Each of the participants outlined below is a unique independent entity, and they each communicate with one another in processes.

Participant 1: Coorest

Coorest is responsible for coordinating all the activities related to the NFTree lifecycle. They are responsible for initiating, managing, and verifying carbon-token transactions. They receive purchase requests from NFTree Customers, validate payment, and carbon-offset data and trigger smart contracts with other participants like the Polygon PoS Blockchain to mint NFTrees. Coorest has contracts with Chainlink Oracle as well to verify satellite data and FileCoin to store metadata. In terms of the choreography, Coorest directs the process and ensures that all required verifications occur prior to registry, and each NFTree is linked to authentic carbon data.

Participant 2: PolygonPOS Blockchain

PolygonPOS Blockchain is responsible for executing transactions within the system. It mints new NFTree tokens and records transaction data in the blockchain. Polygon enables token creation, ownership tracking and transfer, and secure transfer between participants. Transactions are permanently logged, ensuring security, transparency, and authenticity.

Participant 3: FileCoin Storage

FileCoin is responsible for the data storage of the Coorest process, for all data associated with NFTrees. This data can include satellite carbon sequestration data, ownership information, and sales from the tree, among other details. FileCoin returns unique identifiers used in the PolygonPOS and smart contract, linking NFTrees to the data. FileCoin supports data longevity and accessibility.

Participant 4: Chainlink Oracle

Chainlink Oracle is responsible for determining and verifying the link between the blockchain and the real-world impact of the NFTrees. The participant gathers environmental data such as carbon-sinking statistics and satellite data. Once information is verified, Chainlink Oracle provides input to Coorest's smart contract, enabling the automation. In the choreography, Chainlink Oracle is integral in ensuring the accuracy of data dependencies in the process.

Participant 5: Registry Blockchain Smart Contract

The Registry Blockchain Smart Contract is a record of all NFTrees. It keeps track of all trees and tokens, and is linked to the associated filecoin that holds additional information, such as ownership. The registry is a trustworthy information source for all participants.

Participant 6: NFTree Customer

The NFTree customer is the person who purchases the NFTrees and gets the carbon tokens each tree produces. The customer initiates the process when they submit a purchase request through Coorest and provide data such as the desired token and payment information. Post verification and minting, the customer receives an NFTree token as proof of carbon offset. In the choreography, this participant initiates the process.

Participant 7: Carbon Token Buyer

A carbon token buyer is someone trying to purchase carbon tokens from previous owners. This participant must interact with the Polygon POS Blockchain and Registry to execute token transfers. Token buyers create a market in the system, which makes tokens circulate among multiple owners. In the choreography, token buyers are often at the end of the token lifecycle, as they will purchase tokens to settle their own carbon offset or burn them.

2.0 Textual Description of the Choreography

The choreography begins when an NFTree customer submits a project request to Coorest. Coorest then performs a validation check on the data submitted. If the data is invalid, then Coorest sends a message back to the NFTress customer requesting the corrected data. This process implements a “receive pattern with message buffering” where the resubmissions are queued until Coorest’s automated validation is ready to receive them. If validation is successful, Coorest will register the project with the Registry Blockchain Smart Contract using a “send pattern with a binding time”. The message has a call to “register(key, address)” where the key follows a format of “project:PRJ[ID]:owner”. The binding time is runtime since the address is fixed, but the project key is generated once the project passes validation.

Once a project is registered and trees have grown, Coorest then begins the dMRV (data Monitoring, Reporting, and Verification) process to measure carbon capture. This demonstrates a “one-to-many send pattern” since Coorest sends data collection requests to 3 independent sources: Chainlink Oracle for satellite data, a field sensor network, and local reporting agents. These 3 collectors operate independently and implement a “one-from-many receive pattern” since Coorest waits until all 3 sources provide their data before proceeding.

Coorest performs a plot verification process to ensure all plot data is valid. If the data shows any inconsistencies, the process implements a”multi-responses pattern” where Coorest then requests additional data from field sensors until sufficient data is collected to remove any inconsistencies, and then the plot is validated. The complete and validated dataset is then stored in Filecoin Storage. The dMRV Verification process depicts a combination of the service interactions that work together: “one-to-many send” for parallel data collection requests, “one-from-many receive” for the synchronization of all responses, “multi-responses”, and “multi-instance subprocess”.

When the dMRV Verification confirms the carbon capture, Coorest calculates the total verified carbon amount and initiates token generation. This then implements a “relayed request pattern” where Coorest sends a token minting request to the Registry Blockchain Smart Contract. The Registry will validate the request and relay this to Polygon PoS, which mints the carbon credit token and sends confirmation back through the Registry. The Registry then sends a final confirmation back to Coorest. The tokens are registered with the key format “token:CO2-NFT-[TOKEN_ID]:owner”. Coorest publishes the token data and verification report to Filecoin Storage and then notifies the NFTree Customer through the investor portal.

When a Token Buyer wants to either retire or acquire carbon credits, they will submit a request to Coorest. This implements a “racing incoming messages pattern” where the process waits for the first message to arrive between a transfer request or a retirement request. The process path is determined by which message arrives first, since this is a deferred choice. Coorest will verify the token ownership on the Registry Blockchain Smart Contract utilizing “isRegister(tokenKey)”. If it is a transfer request, Coorest will utilize “transfer(tokenKey, newOwnerAddress) on the Registry. This updates the ownership record and causes a transfer on Polygon PoS to move the token to the new owner’s wallet. If it is a retirement request, Coorest utilizes “unregister(tokenKey) on the Registry. The Registry will mark the token as retired and tell Polygon PoS to burn the token. A retirement certificate is then issued to the buyer.

3.0 Choreography Diagram

The images below show the High-Level Structure Model and the High-Level Behavioural Model created in Visio.

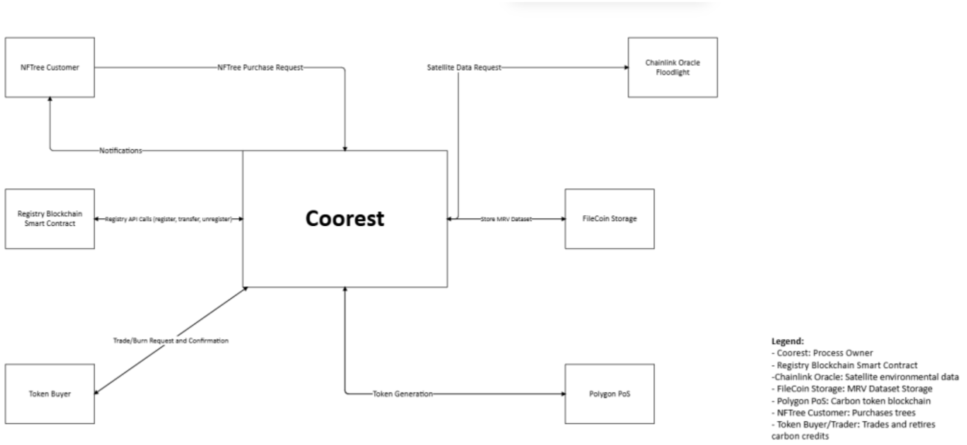
NFTree Customer
NFTree Purchase Request
Satellite Data Request
Chainlink Oracle Floodlight
Notifications
Registry Blockchain Smart Contract
Registry API Calls (register, transfer, unregister)
Coorest
Store MRV Dataset
FileCoin Storage
Trade/Burn Request and Confirmation
Token Buyer
Token Generation
Polygon PoS
Legend:
- Coorest: Process Owner
- Registry Blockchain Smart Contract
-Chainlink Oracle: Satellite environmental data
- FileCoin Storage: MRV Dataset Storage
- Polygon PoS: Carbon token blockchain
- NFTree Customer: Purchases trees
- Token Buyer/Trader: Trades and retires carbon credits

Figure 1: High-Level Structure Model

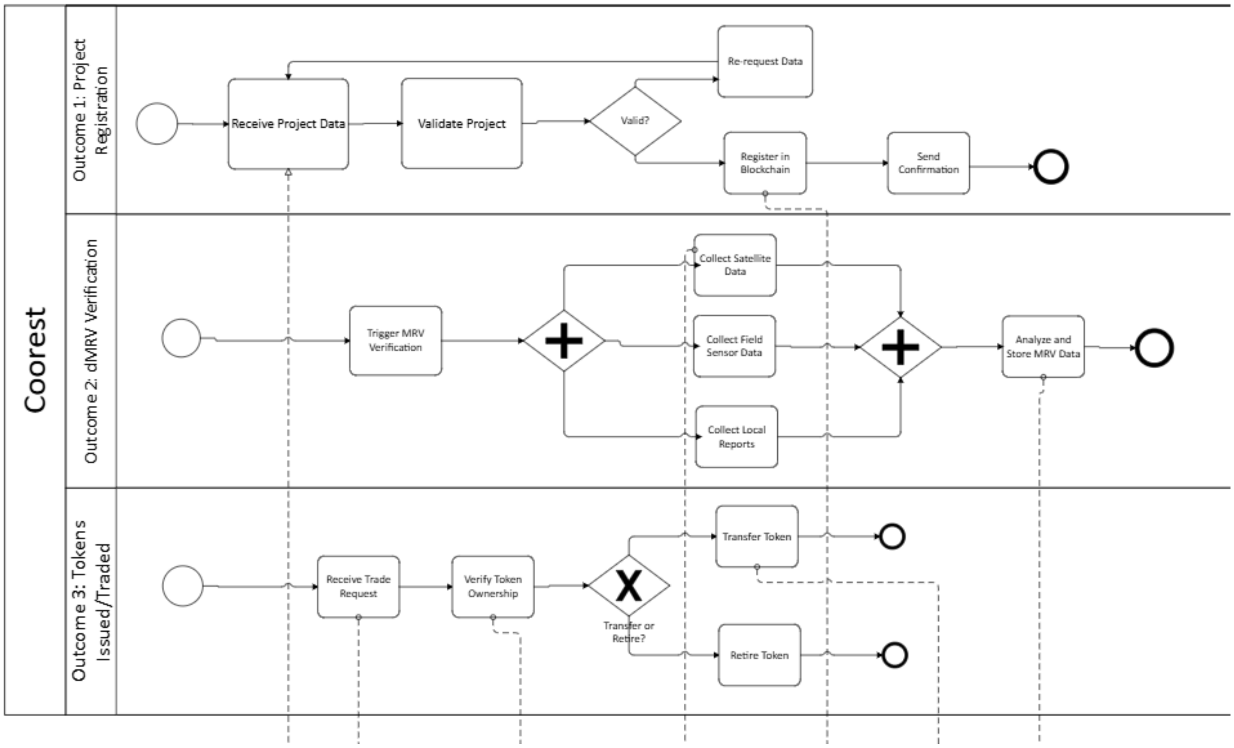

Figure 2. High-Level Behavioral Model

4.0 Resource Classification Description

The resources classification can be broken down into three main groups: organizational, external, and software resources. Organizational resources are those that perform internal business operations, external resources are independent systems that occur outside of the main business and software resources are technical items that enable interactions between organizational and external resources. The table below shows the breakdown of each of the resources into its respective category and sublane/pool.

Resource Classification of Coorest		
Organizational Resources	*External Resources*	*Software Resources*
Main Pool: Coorest	Pool 1: Chainlink Oracle	API Registry Package
Sublane 1: Registry Management	Pool 2: FileCoin Storage	
Sublane 2: dMRV	Pool 3: Polygon POS Blockchain	
Sublane 3: Token & Issuance		

5.0 Activity Table

The activity table detailing the events of the BPMN Choreography created in Visio are outlined below.

Activity Name	**Description**	**Resources**	**Data Object**
Receive Project Proposal	Receive Project Submission	Coorest	Project Proposal
Validate Project Data	Check if project data is valid. If invalid, return back to be processed	Coorest	Automated Validation System
Re-request Data	Sends message to customer requesting the corrected submission data	Coorest	
Call Registry: register()	Store project record on	Registry Blockchain	

	blockchain	Smart Contract	
Collect Satellite Data	Request environmental data from Chainlink Oracle	Coorest	Satellite Data
Collect Field Sensor Data	Gather MRV Data	Coorest	Field Sensor Data
Collect Local Reports	Collect data from local reports	Coorest	Local Data
Aggregate and Analyze MRV Data	Combine data from sensors and satellite to compute carbon metrics. If the data is invalid, loop back to data collection	Coorest	
Plot Verification	Verification subprocess for individual plot	Coorest	
Compute per Plot Carbon Amount	Calculate carbon capture for the verified plot	Coorest	
Return to collect field sensor data	If plot is inconsistent, request more data	Coorest	
Compose Verified MRV Report	Create a report of the final verified data for token issuance	Coorest	
Store MRV Dataset	Upload data to Filecoin for storage	Filecoin	MRV dataset
Relay Request to Polygon PoS	Forward minting request from Registry to blockchain	Registry Blockchain Smart Contract	
Generate Carbon Tokens	Generate tokens on Polygon PoS based on the verified MRV report	Polygon PoS	Carbon Tokens
Notify Investor Portal	Send a notification to NFTree	Coorest	

	customer of token generation		
Verify Token Ownership	Confirm current owner matches blockchain records	Registry Blockchain Smart Contract	Carbon Tokens
Transfer Carbon Tokens	Transfer tokens to new owner using transfer(tokenKey, newOwner)	Polygon PoS, Registry Blockchain Smart Contract	Carbon Tokens
Retire Carbon Tokens	Unregister the tokens when the carbon offset is retired using unregister(tokenKey)	Polygon PoS, Registry Blockchain Smart Contract	
Record Retirement Proof	Document the proof of carbon token retirement in Coorest's system	Coorest	Retirement Proof

6.0 Registry Blockchain Smart Contract Justification

In the Coorest choreography, a blockchain is used to keep all verified tree and token data records immutable and accessible to all participants involved. The blockchain supports operations like register(), transfer(), unregister(), and ensures all ownership, transactions and verifications across the system. Each of the operations in the blockchain makes the system synchronized and secure. The blockchain anchors the registry and keeps a truth ledger connecting the systems together. The setup also prevents duplication of carbon assets and provides a clear transaction log for all carbon tokens. Without the blockchain registry to support component coordination, the operation would not run as seamlessly.