# Blockchain for Carbon Trading


**Group 6:**
**Ernest Chng Yong Cheng**
**Jordan Seow Hua Jun**
**Jolene Tan Shimin**
**Kaustubh Jagtap**
**Shervin Lee Jia Ning**

# Our Team

Kaustubh Jagtap

Ernest Chng

Jordan Seow

Jolene Tan

Shervin Lee

# Our Agenda

01 Problem Statement

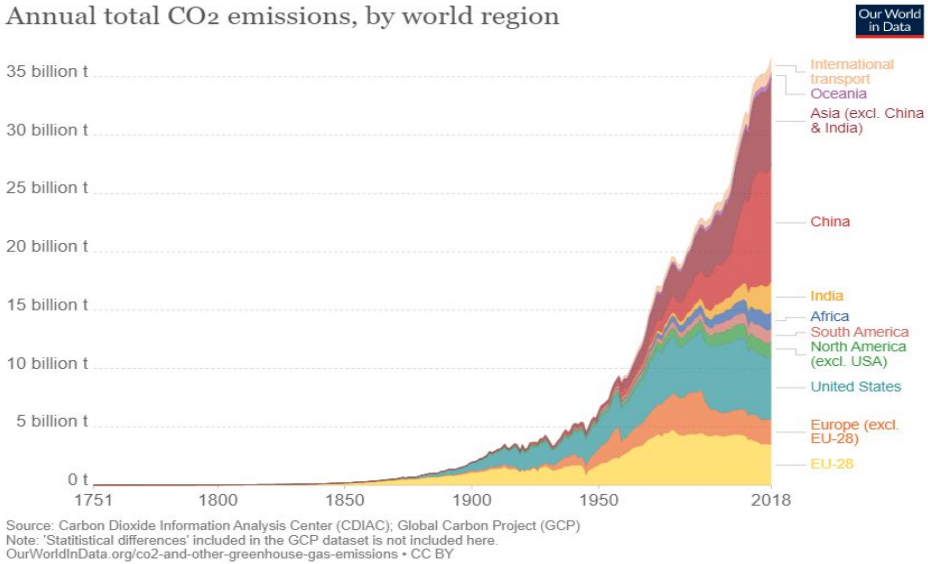
02 Proposal Overview

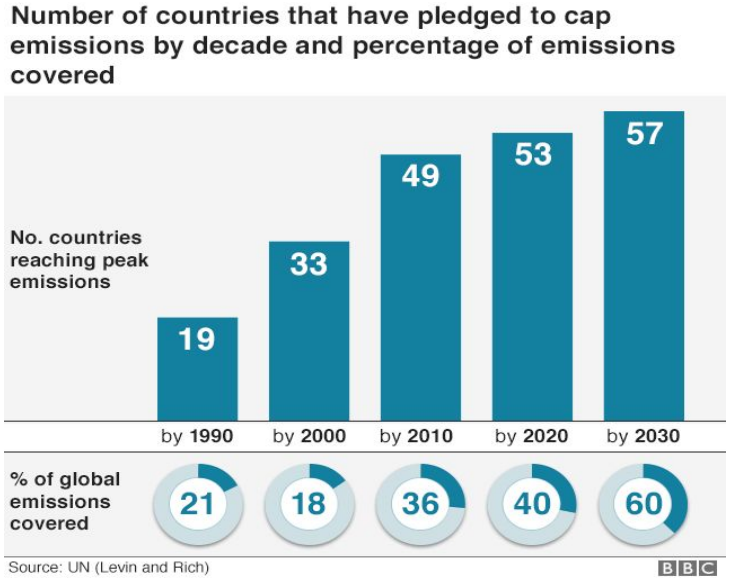
03 Back-End Framework

04 Front-End Demo

05 Limitations and Extensions

# Problem Statement




Greenhouse gas emissions are increasing, and they are harming the environment.

Governments want to address this.
But How?

# Problem Statement

## *3 Methods to Reduce Greenhouse Gas Emissions*

**Setting a specific limit**

Companies will not be allowed to exceed the specified limit

**Carbon taxation**

Introduce a carbon tax where the company pays for the amount of $CO_2$ produced

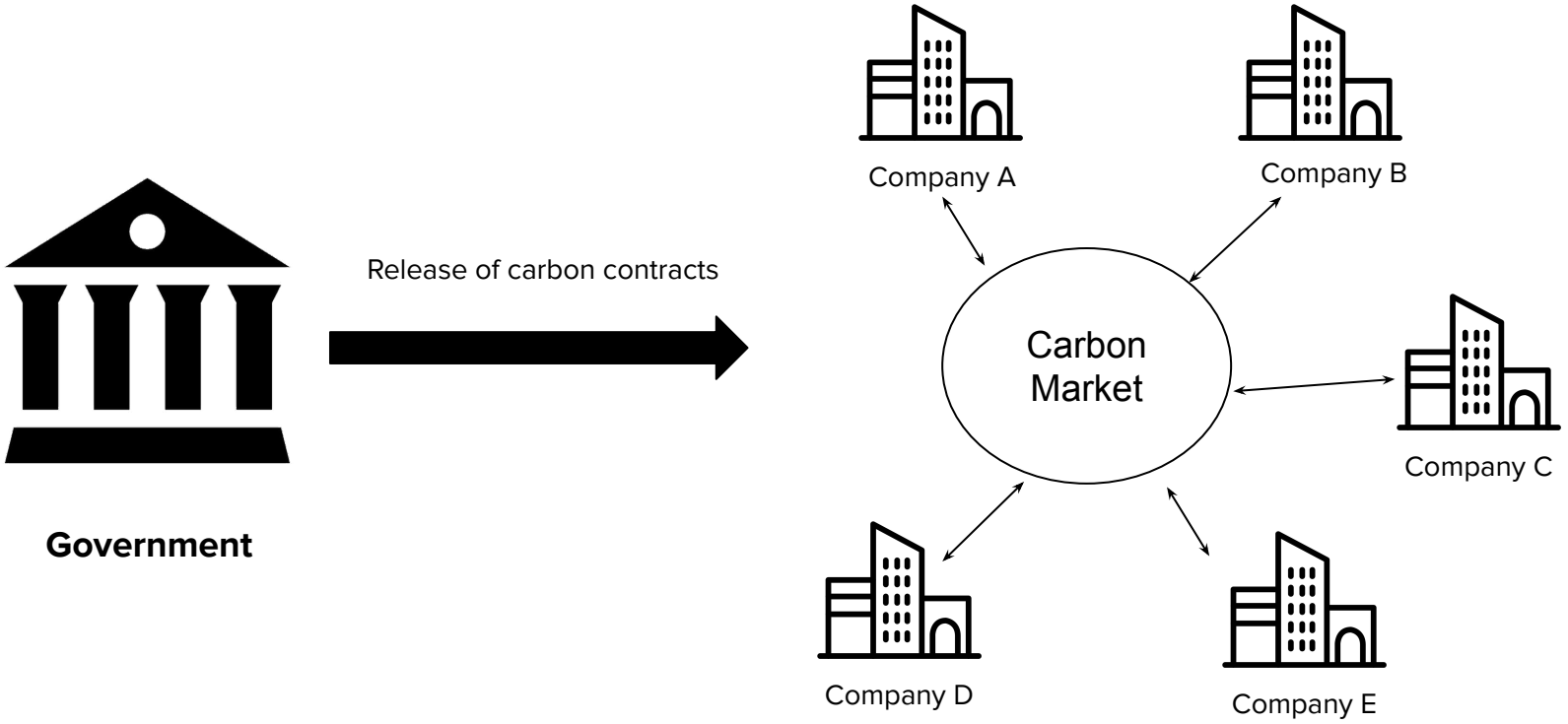
**Carbon market**

Implement an emission cap-and-trade scheme.

Doing this on the blockchain has been talked about a lot; this is our proposed solution.

# Problem Statement

## *How does the Cap-And-Trade Scheme work?*



# Solution

A carbon token system that is based off the ERC20 ethereum standard. By moving this to the blockchain, we eliminate the margins lost to liquidity providers, allow for better transparency and increase participation in the carbon market and awareness of the green agenda. Specifically, these are the ways in which various stakeholders benefit:

## Governments

1) High transparency leads to better oversight of carbon market.

2) Lower cost of enforcement due to immutability and single history, reducing the need for unnecessary lawsuits that may otherwise result from ambiguity.

## Businesses

1) Get rewarded and recognized from going green. Strong incentive for green Capex.

2) Market based on blockchain has greater efficiency, thereby saving trading costs to liquidity providers.

## Society

1) Get to participate in and monetize personal carbon reduction efforts.

2) Incentive structure for third-party verifiers moves away from bias towards corporates


Back-End

# Key Functionalities

## Generators

- Wind Farms
- Tree-planting projects
- Solar Panel owners

➜ Tokens will be minted to this group

## Consumers

- Carbon emitters
- Corporates & Businesses

➜ They hold carbon tokens (which represent credits up to which they can emit)

➜ Allowed to buy & sell (trade) tokens.

## Validators

Accredited, technically competent consultants to check emissions

➜ There are many accredited validators in the system who can cross-check each other's work, which incentivizes them to act in the best interest of the system.

## Regulators

Government-linked regulatory bodies to provide oversight and ensure compliance

➜ Regulators own the contracts in the system.

**Coins:**

➜ Represent Carbon Token (license to emit carbon).
➜ Credits can traded on the blockchain by consumers and generators, which acts as our carbon exchange.

# Carbon Credit

## Life-Cycle of Carbon Credit

Ramesh, a registered Generator, plants a tree and sends a photograph to the network.

His project gets approved and he gets awarded 0.5 tokens.

Chad, an entrepreneur, sets up a construction company. Based on a strict assessment of his emission needs, his company get awarded 150 tokens upon inception.

| Main Functions | Purpose |
|---|---|
| ***createGenerator***(<br>*uint generatorId*<br>) *isRegulator* | Regulator creates a Generator that is assigned a unique generator ID |
| ***createConsumer***(<br>*uint consumerId*<br>) *isRegulator* | Regulator creates a Consumer that is assigned a unique consumer ID |
| ***allocateCredit***(<br>*uint generatorId,*<br>*uint credit*<br>) *isValidators* | Regulator allocates carbon credits to a specific generator |
| ***reportEmission***(<br>*uint consumerId,*<br>*uint emission*<br>) *isValidator* | Validators report the amount of actual emissions a consumer has emitted and will be stored in the struct Consumer |
| ***checkEmission***()<br>*isRegulator*<br>*returns (uint[] violators)* | Regulators check if consumers have violated the credit limit and returns a list of the violators by consumer ID |

# Carbon Credit

## Life-Cycle of Carbon Credit

Ramesh can then list the tokens he has generated on the marketplace, to monetize his work.

Chad, upon realizing that he may need a licence to emit more (i.e. say 20 more tokens, buys this from the marketplace.

Next year, he is incentivised to undertake investments in clean cement processors to reduce emissions.

| Main Functions | Purpose |
|---|---|
| ***createGenerator***(<br>*uint generatorId*<br>) *isRegulator* | Regulator creates a Generator that is assigned a unique generator ID |
| ***createConsumer***(<br>*uint consumerId*<br>) *isRegulator* | Regulator creates a Consumer that is assigned a unique consumer ID |
| ***allocateCredit***(<br>*uint generatorId,*<br>*uint credit*<br>) *isValidators* | Regulator allocates carbon credits to a specific generator |
| ***reportEmission***(<br>*uint consumerId,*<br>*uint emission*<br>) *isValidator* | Validators report the amount of actual emissions a consumer has emitted and will be stored in the struct Consumer |
| ***checkEmission***()<br>*isRegulator*<br>*returns (uint[] violators)* | Regulators check if consumers have violated the credit limit and returns a list of the violators by consumer ID |

# MarketPlace

## Listing Tokens

Next year, after Chad's investment in Clean Technology pays off, he has an excess of 50 tokens.

He lists these on the marketplace, at a price he feels comfortable accepting, say 1400.

## Listed Tokens on the Marketplace

| Quantity | Price |
|---|---|
| 30 | 1100 |
| 20 | 1350 |
| 50 | 1400 |
| 75 | 1480 |
| 15 | 1600 |

# MarketPlace

## Buying Tokens

Chad's cousin Brad, an irresponsible business owner who has not focussed on 'greening' his company, realises he needs **80** tokens to cover for the year's carbon emissions.

He sends a buy order to the market.

## How The Order Is Filled

Complete Fill

Complete Fill

Partial Fill

| Quantity | Price |
|---|---|
| 30 | 1100 |
| 20 | 1350 |
| 50 | 1400 |
| 75 | 1480 |
| 15 | 1600 |

# MarketPlace

## Buying Tokens

Chad’s cousin Brad, an irresponsible business owner who has not focussed on ‘greening’ his company, realises he needs **80** tokens to cover for the year’s carbon emissions.

He sends a buy order to the market.

## New Market State

| Quantity | Price |
|---|---|
| 20 | 1400 |
| 75 | 1480 |
| 15 | 1600 |


Front-End
Demo

# Limitations & Extensions

# Limitations

## Governance and Trust

- Network not fully decentralised
- Requires trust in government and integrity of credits issued
- Unequal trust between players in a global model

## Enforcement

- Real world: Consumers forced to participate through legislation
- Not much incentive to participate without laws

## Verification

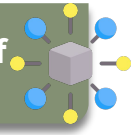
- High complexity of verification and accreditation
- Need more complex technology

# Extensions

## Global trade of carbon credits

- Our model can be implemented on a global scale
- Requires cross-border cooperation
- On-chain governance design

## Full decentralisation of carbon market



- When taken on a global scale, cross-border trust in governments becomes an issue
- Need to be fully decentralised to minimise risks

## Automated Measurement, Reporting and Verification (MRV)

- Aim to decrease MRV time and cost
- Complementary technologies:
  - Computer vision for project verification
  - Earth observation satellites to report emissions


THANK YOU