Smart Gambling

Ambreen Simon, Sebastien Vezina, Stephen Chen,
Wazarat Hussain, & Val De Franco

Background

Growing interest in adopting blockchain for online gambling:

- Privacy protection
- Data security
- Ease of virtual transactions
- Circumvent local regulations and restrictions

Aim

To create a simple online game utilizing:

- Smart contract
- Crypto-currency such as Ether for betting
- GUI (frontend)

Via GUI, a player will bet their wei and roll a virtual die.

- Winners receive 5x the bet + refund of the original bet.

Resources

- Solidity
- OpenZeppelin, SafeMath
- ChainLink
- Web3.js - Ethereum JavaScript API
- Remix, MetaMask, Rinkeby Ethereum Testnet
- HTML, JavaScript

Chainlink

METAMASK

Points of Fascination

Challenge:

Random Numbers

Solution:

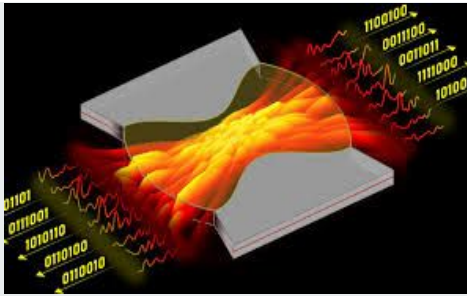
Oracle = Chainlink + Verifiable Random Function

What is an Oracle?

What is an Oracle?

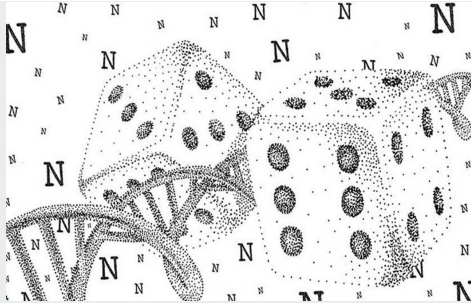
- Sends data from the outside world to the blockchain
- Smart contracts aren't designed to take care of everything
- Just like normal businesses, sometimes rely on suppliers or vendors, in this case, an Oracle such as Chainlink.

Random Numbers



- Random ≠ Random
  - Algorithms use a seed such as time or mouse movements.
  - Can be solved.
  - Usually random enough for most applications.

Random Numbers



Alternatives:

1. Use Blockhash.
2. Use centralized API or Oracle.
3. Verifiable Random Function = VRF (Chainlink).

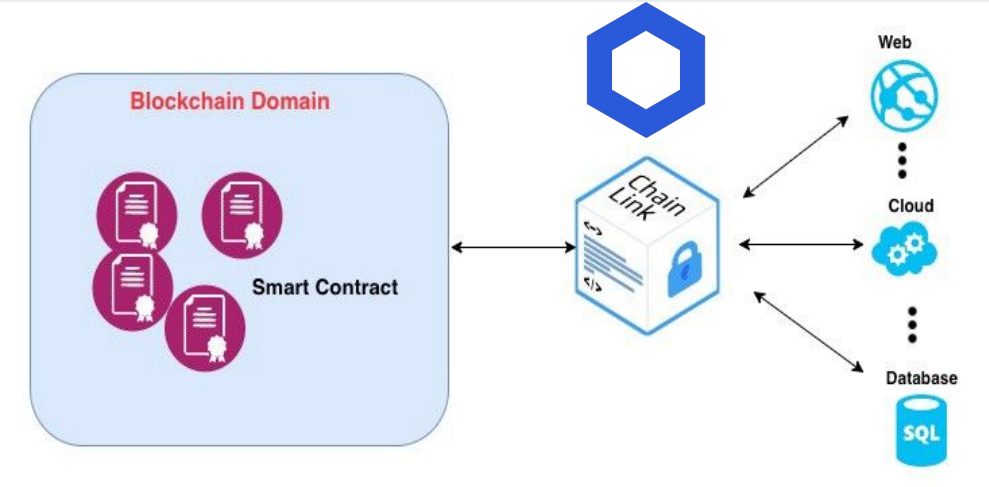
Random Numbers

ALTERNATIVES	CRITERIA		
	Decentralized	Really Random	Miner Conflict of Interest
BlockHash	✓	✓	X
Centralized Oracle	X	?	✓
Chainlink / VRF	✓	✓	✓

Random Numbers

Solution: Chainlink/VRF.

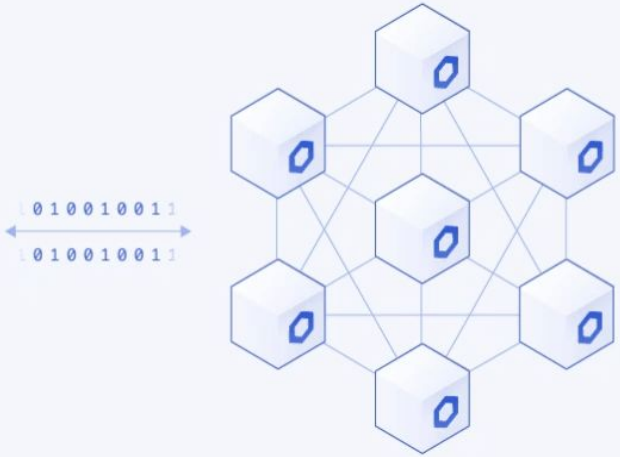
- As a blockchain, multiple nodes operating.
- Removes single point of failure.
- VRF is usable for our project.
- Specifically, we use Chainlink.



Chainlink (LINK)

Chainlink (LINK)

VRF



Smart contracts

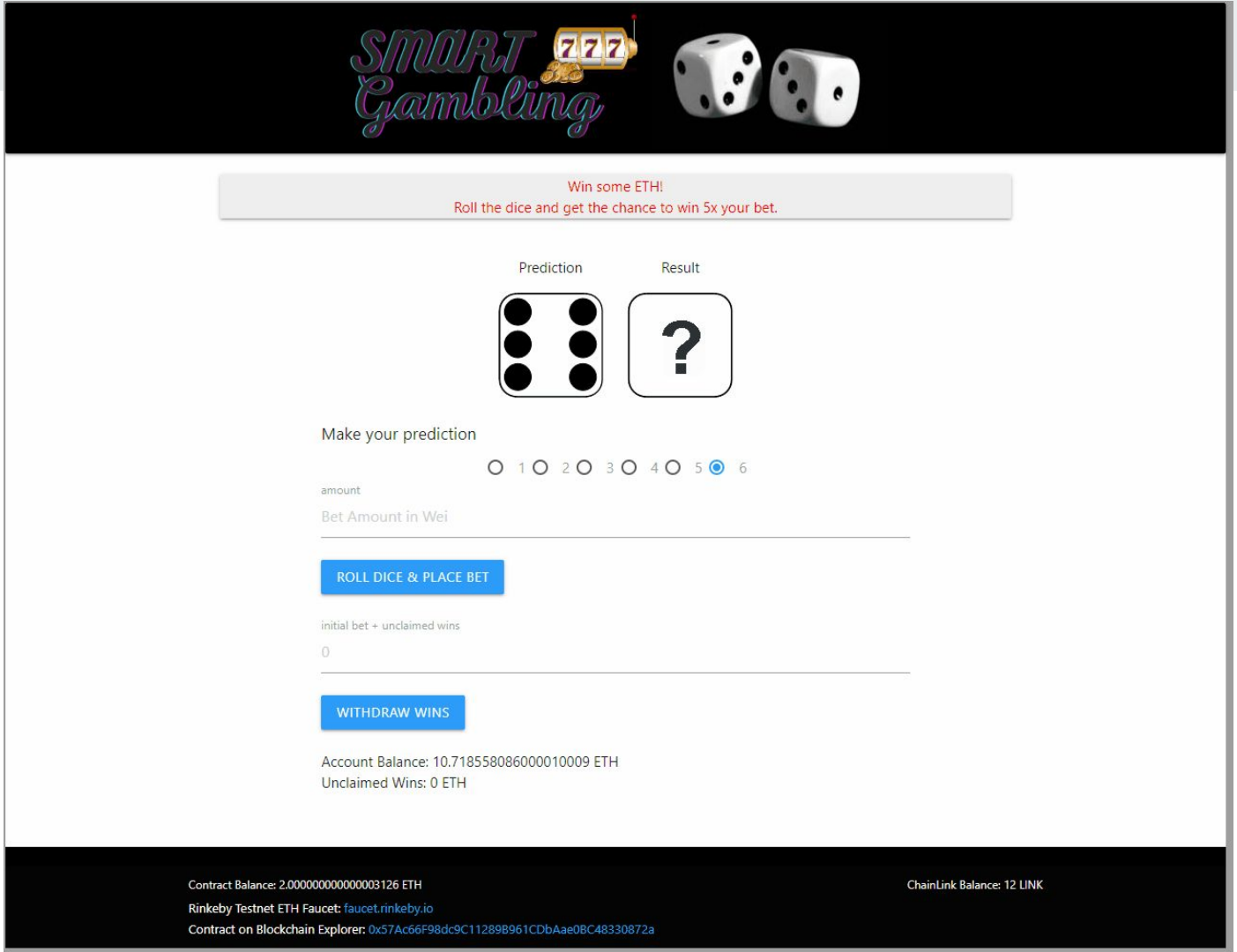
Events

- Bridge between contract & front-end.
- We are using 3 events: BetPlacedEvent, BetResultEvent, withdrawWinsEvent.
- Front-end monitors for these events to refresh GUI.

Players

- Supports multiple players: use player address and timestamp as unique identifier.
- Code must account for:
  - deposits
  - amount owing to players
  - unresolved bets in queue
  - withdrawals

Demo


SMART Gambling
Win some ETH!
Roll the dice and get the chance to win 5x your bet.
Prediction
Result
?
Make your prediction
1 2 3 4 5 6
amount
Bet Amount in Wei
ROLL DICE & PLACE BET
initial bet + unclaimed wins
0
WITHDRAW WINS
Account Balance: 10.718558086000010009 ETH
Unclaimed Wins: 0 ETH
Contract Balance: 2.000000000000003126 ETH
ChainLink Balance: 12 LINK
Rinkeby Testnet ETH Faucet: faucet.rinkeby.io
Contract on Blockchain Explorer: 0x57Ac66F98dc9C11289B961CDbAae0BC48330872a

Future Directions

Backend:

- Automatic refill of Link tokens on UniSwap or other platform
- Determine real world value of Link in order to determine minimum bets.

Frontend:

- More/different games.
- Implementing interactive graphics.
- Multi-language interface.
- Provide log/audit of player’s bets.

Questions?

Aim

To create a simple online gaming interface utilizing smart contract and cryptocurrency such as Ether as the currency for betting.

Via a web interface using underlying smart contract, a player will bet any amount of wei and roll a virtual dice. If the user correctly guesses the number, s/he will win 5x the amount back in addition to being refunded her/his initial bet.

To be deleted

Collaboration With Prophet.finance — Decentralized Lottery Game Using Chainlink VRF Oracles | by trinityprotocol | Medium

PROPHET PRESENTS: THE OFFERTORY. offertory | by Prophet | Medium

Ambreen Simon, Sebastien Vezina, Stephen Chen,
Wazarat Hussain, & Val De Franco

Resources

- Solidity
- OpenZeppelin, SafeMath
- ChainLink
- Web3.js - Ethereum JavaScript API
- Remix, MetaMask, Rinkeby Ethereum Testnet
- HTML, JavaScript

Smart contract

The core of our smart contract to is generate a random number from 1 to 6.

Issues:

1. Players can potentially figure internal random number generator.
2. Miners can choose not to confirm contract if s/he is betting and the result is not favorable.

Solution:

Use a Verifiable Random Function (VRF) from the decentralized Oracle (ChainLink) to instil confidence in game participants that the game results are truly random.

What is an Oracle?

An “oracle” sends data from the outside world, such as the daily temperature or the number of votes a political candidate received, to a blockchain such as Ethereum. A smart contract on the blockchain can then use the data, typically to make a decision about whether to dispense money and to whom.

Why they are used?

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management.

Centralized vs Decentralized?

As its name implies, decentralized systems don't have one central owner. Instead, they use multiple central owners, each of which usually stores a copy of the resources users can access. A decentralized system can be

Random Numbers

- Block Hash:
  - Potential conflict of interest with miners
  - If miners are betting, might not confirm losing bets and only confirm winning bets.
  - How can a non-mining player be sure?
  - Not Usable for our application.

Random Numbers

- Centralized, External API or Oracle.
  - How can a player be sure that the outside source is truly generating a random #?
  - What if the sight goes down? Or corrupted?
  - Goes against principle of decentralization.
  - Not Usable for our application.

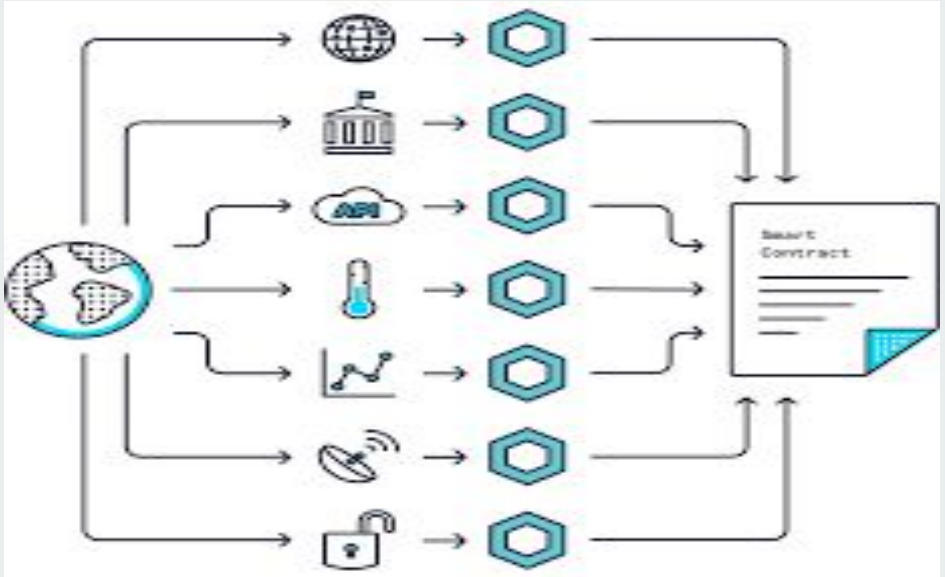
Random Numbers

- Solution: VRF.
  - Utilizes Chainlink = a blockchain.
  - Random # is verified/confirmed within the Chainlink blockchain.
  - So is decentralized, **not controlled**.

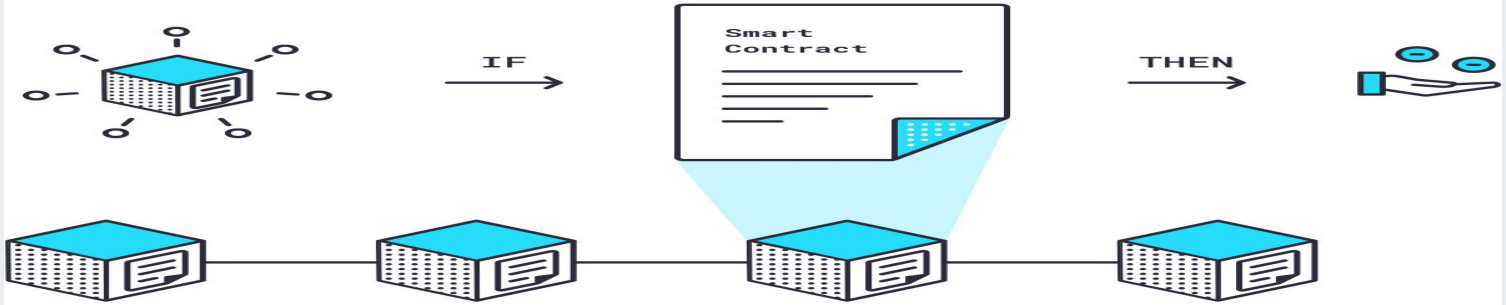
What is an Oracle?

- Sends data from the outside world to the blockchain.
- Smart contracts aren't designed to take care of everything.
- Just like normal businesses, sometimes rely on suppliers or vendors, in this case, an Oracle such as Chainlink.

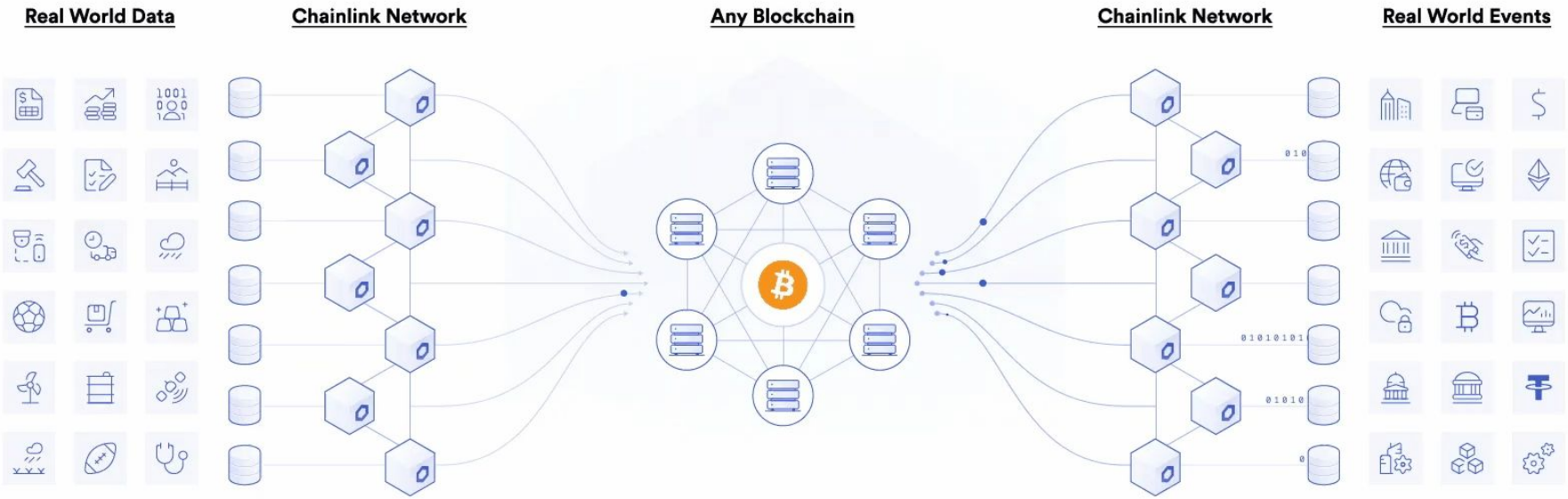
Chainlink Contracts

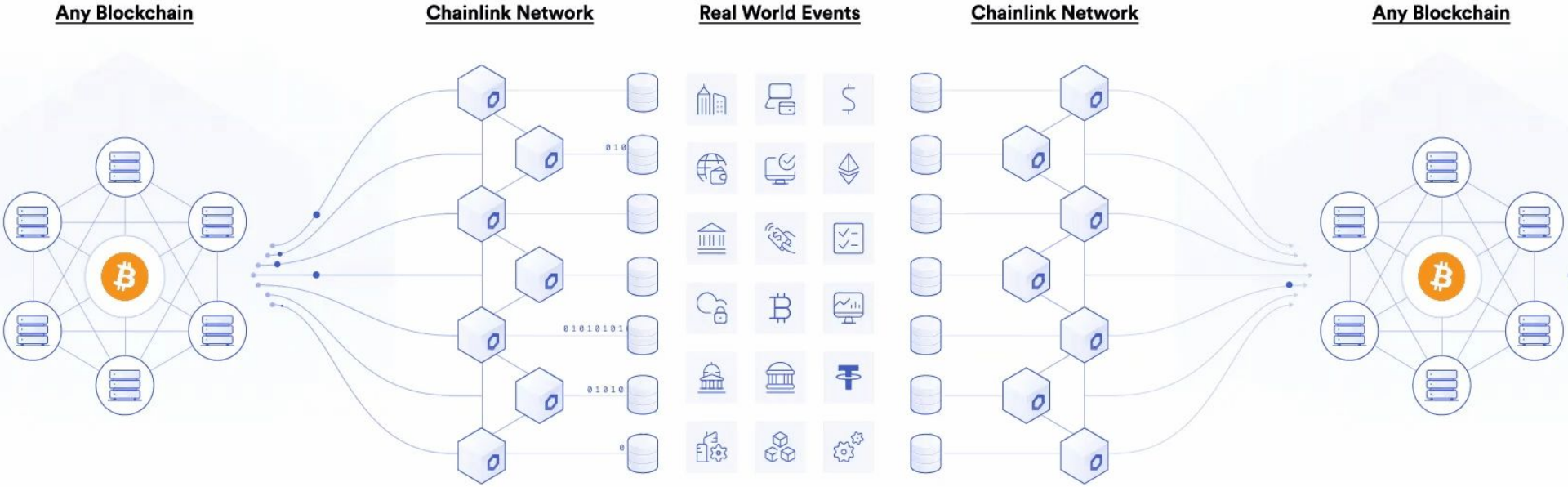


The benefits of Chain Link and how it functions



- Smart contracts are pre-specified agreements on the blockchain.
- It evaluates information and automatically execute when certain conditions are met.
- Crowdfunding is a good example: if a certain amount of Ether is deposited into a smart contract by a certain date, then payment will be released to the fundraiser


Real World Data
Chainlink Network
Any Blockchain
Chainlink Network
Real World Events
010
01010101
01010
0


Any Blockchain
Chainlink Network
Real World Events
Chainlink Network
Any Blockchain
010
01010101
01010
0