



Chainlink

# MetaBrain NFT



**Randomize on-chain data with Chainlink VRF**

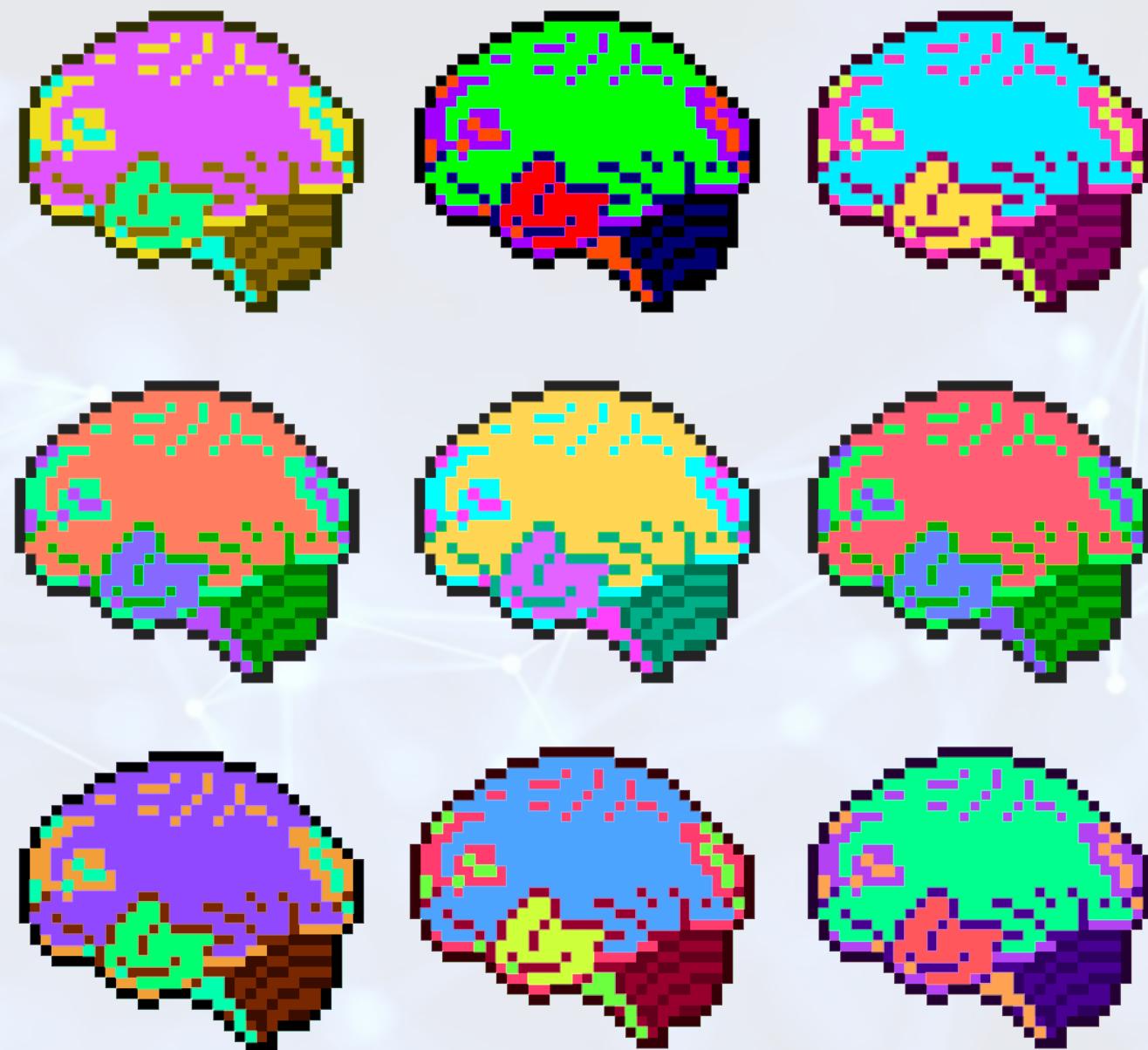
# Using Chainlink VRF for NFT

- On Chain Randomness
- Use Cases
- What is Chainlink VRF
- Randomize NFT Data
- Code Demo: MetaBrain NFT



# Goal: Random MetaBrain NFT

- 1. User mints MetaBrain NFT**
- 2. Generate random brain**
- 3. Each brain has unique traits**
- 4. Provably fair and random**



# What is NFT (ERC-721)?

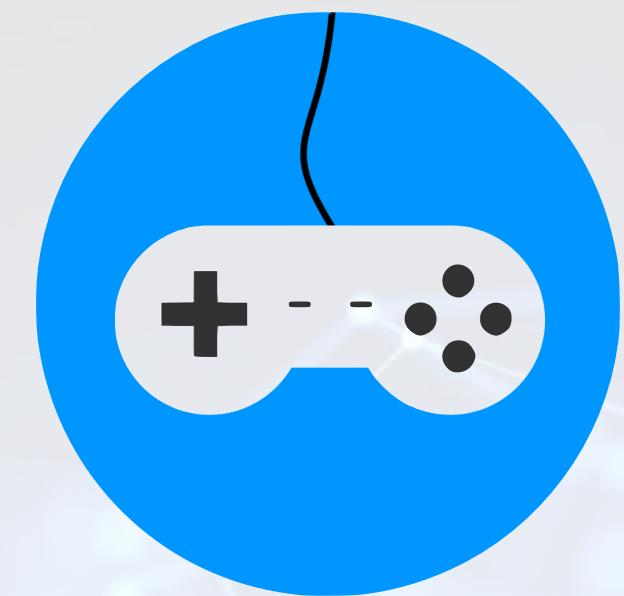
- Non-Fungible Token
- Unique & Non-interchangeable
- Metadata - Attributes, Traits, Rarities
- Live on Blockchain
- Other NFTs: ERC1155 (FT&NFTs)



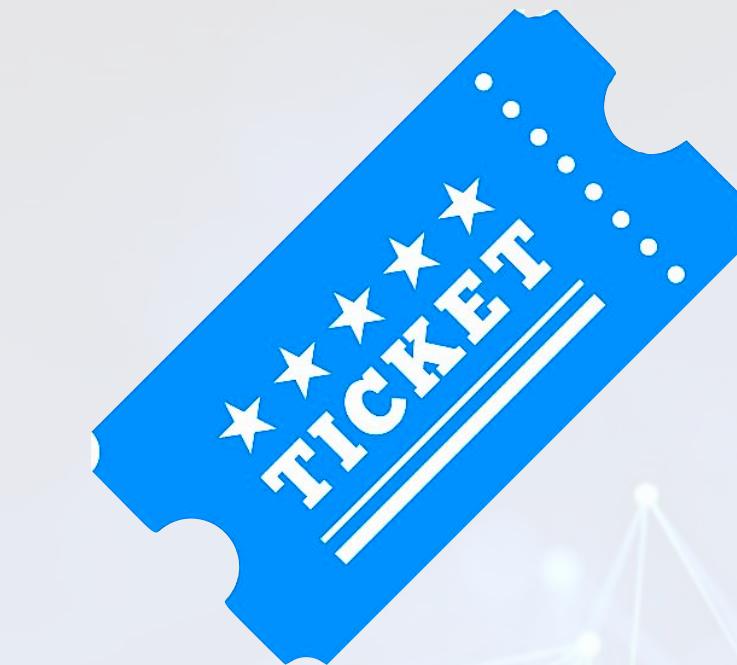
# Randomness Use Cases



- **Unique Traits**
- **Fair Rarity**
- **Random Distribution**



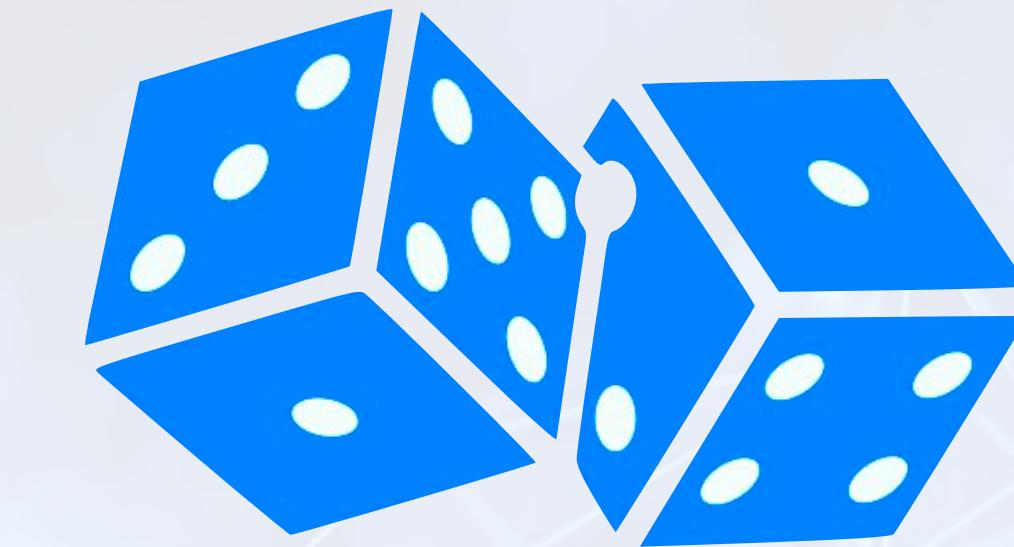
- **Random Outcomes**
- **Game Rewards**
- **Fair Selection**



- **Raffle**
- **Sample Selection**
- **Airdrops/IDOs**

# On Chain Randomness

- **Provably Fair**
- **Provably Random**
- **Cost Effective**
- **Reliable and Secure**
- **Easy to Integrate**



# Challenge of Randomness

- **Blockchain is Deterministic**
- **Decentralized Network**
- **Choose a Trusted Source**

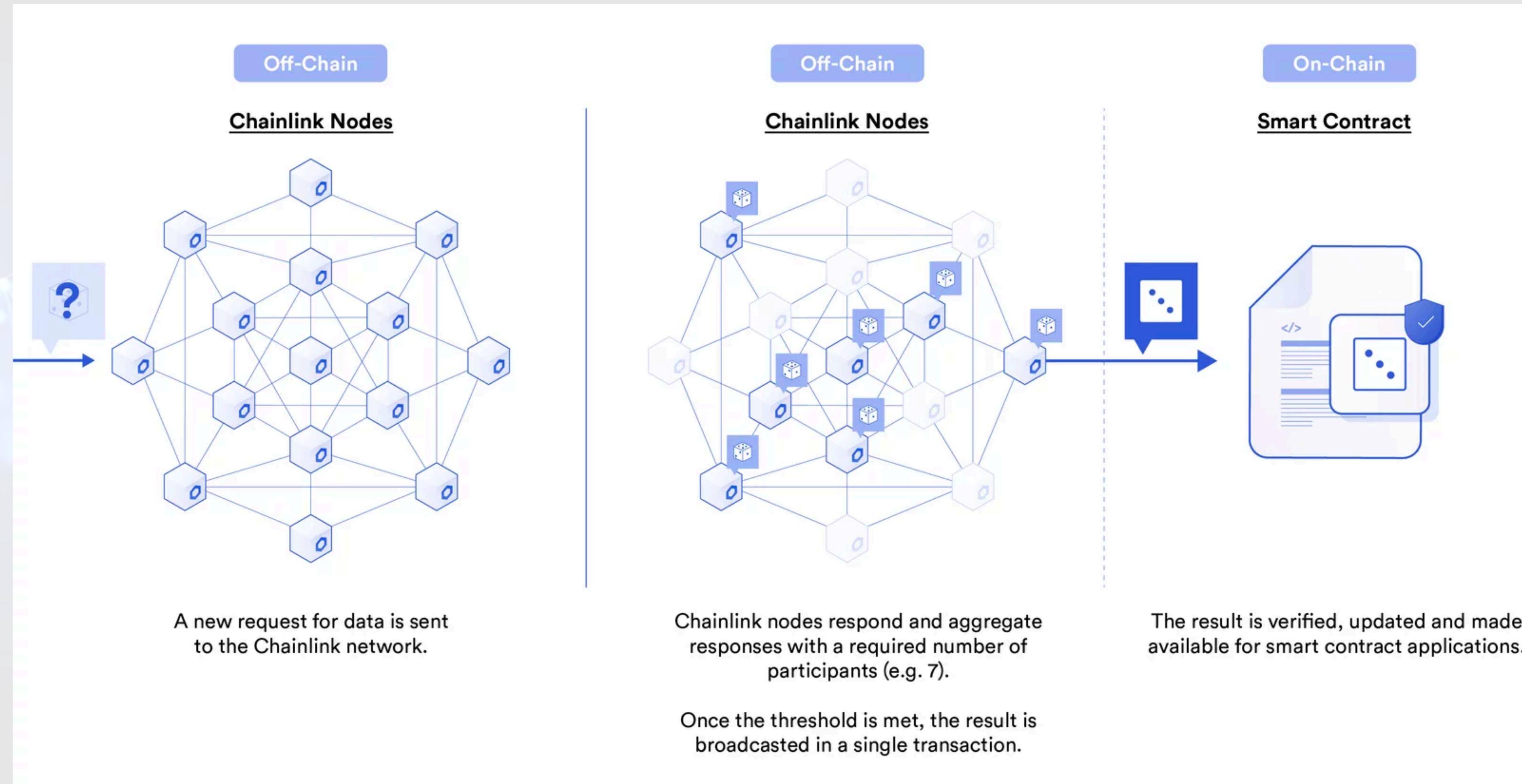


# What is Chainlink VRF?

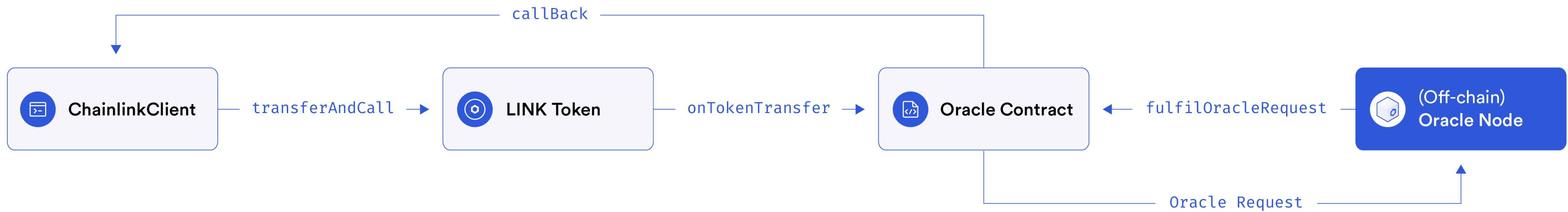
**Chainlink VRF (Verifiable Random Function) is a provably-fair and verifiable source of randomness designed for smart contracts.**



# Chainlink Basic Architecture



# Chainlink Basic Request Model



**Request -> Receive  
(Async Data Cycle)**

# Implement Chainlink VRF

## Two Steps (Transactions)

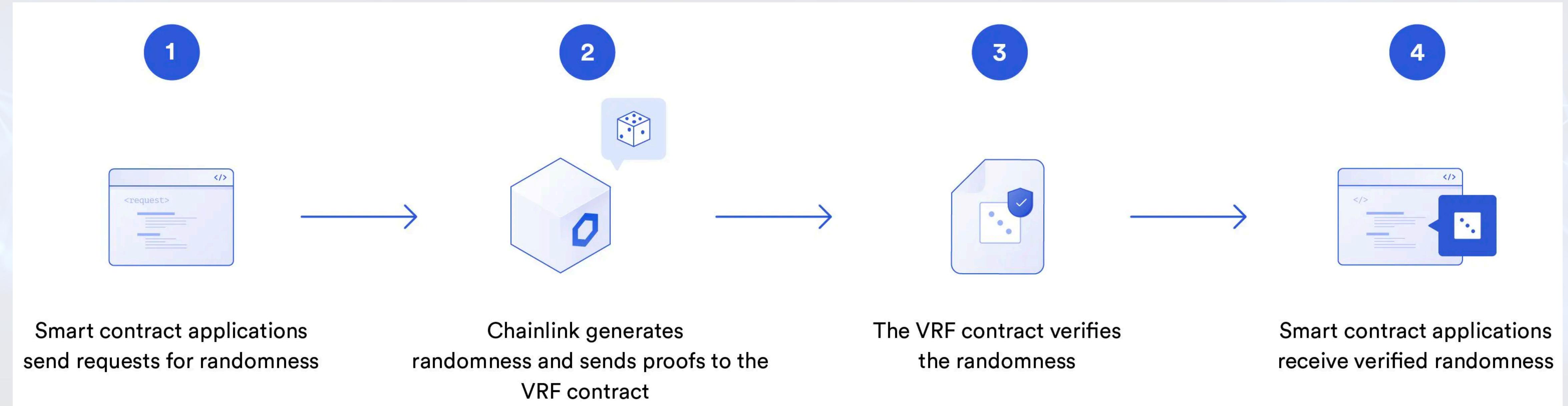
1. Request Randomness
2. Fulfill Randomness (callback, <= 200k gas)

## Components

- **LINK Token** - LINK token address
- **VRF Coordinator** - a contract verifies randomness
- **Key Hash** - public key against the randomness
- **Fee** - require to fulfill a VRF request (~0.1 LINK)



# How Chainlink VRF Works



# Code Demo: MetaBrain NFT

# Let's Code!

## Github

- <https://github.com/reddqian/MetaBrainVRFNFT/>

## Documentation

- <https://docs.chain.link/docs/get-a-random-number/>
- <https://docs.chain.link/docs/vrf-contracts/>

## Blog

- <https://blog.chain.link/create-dynamic-nfts-using-chainlink-oracles/>