# The Oracle Problem

The oracle problem revolves around a very simple limitation—blockchains cannot pull in data from or push data out to any external system as built-in functionality. As such, blockchains are isolated networks, akin to a computer with no Internet connection. The isolation of a blockchain is the precise property that makes it extremely secure and reliable, as the network only needs to form consensus on a very basic set of binary (true/false) questions using data already stored inside of its ledger. These questions include: did the public key holder sign the transaction with their corresponding private key, does the public address have enough funds to cover its transaction, and is the type of transaction valid within the particular smart contract? The very narrow focus of blockchain consensus is why smart contracts are referred to as being deterministic— they execute exactly as written with a much higher degree of certainty than traditional systems.

However, for smart contracts to realize upwards of 90% of their potential use cases, they must connect to the outside world. For example, financial smart contracts need market information to determine settlements, insurance smart contracts need IoT and web data to make decisions on policy payouts, trade finance contracts need trade documents and digital signatures to know when to release payments, and many smart contracts want to settle in fiat currency on a traditional payment network. None of this information is inherently generated within the blockchain, nor are these traditional services directly accessible.

Bridging the connection between the blockchain (on-chain) and the outside world (off-chain) requires an additional and separate piece of infrastructure known as an oracle.

You Got The Gift, But It Looks Like You're Waiting For Something.
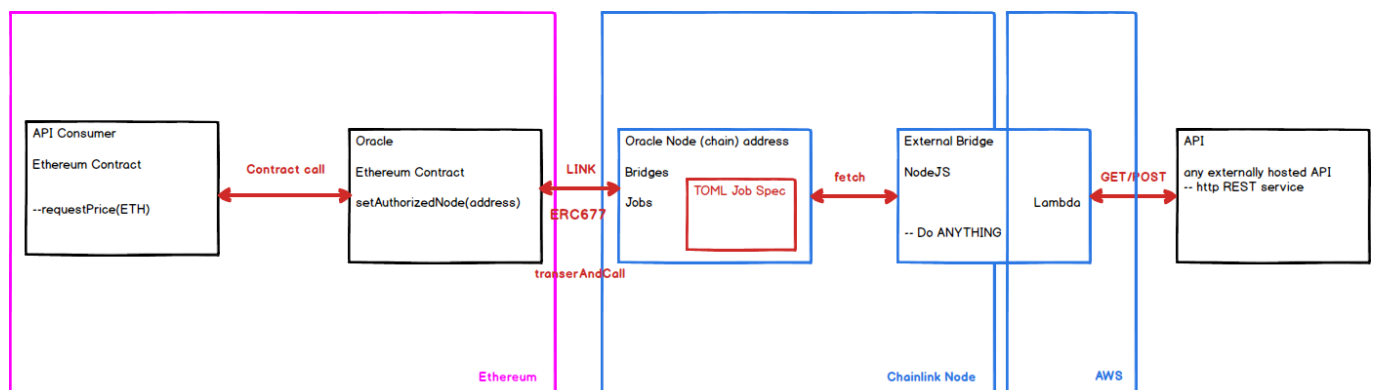
Yes, that Oracle. What this does:

- ✅ Oracle contracts can request employee eligibility and AML checks against external APIs
- ✅ Ex: to collect a vested crypto award, winner must still be an employee
- ✅ Ex: the wallet address used must successfully pass on-chain AML check
- ✅ Resolved via Chainlink Node, with external adapter written in Node.js making the API call to mock services deployed to AWS API gateaway
- ✅ Contracts get updated with API response within the same block transaction.

Ethereum

- ☑ APIConsumer is the contract that will call the Oracle
- ☑ Full request -> fulfill roundtrip working via Chainlink Oracle node

Chainlink

- ☑ Running a **local Chainlink Node in docker**
- ☑ Chainlink Node Jobs, Nodes and Bridges are replicated in toml files in /config
- ☑ External Adapter setup in Node.js
- ☐ **Need to setup distributed deployment of multiple nodes, to light up the oracle pattern**

AWS

- ☑ Lambda deployment for the extenal adpater
- ☑ API Gateway deployment of mock HR and AML services