The Need for Oracles

Smart contracts are significant at immutable storage and verifiable transaction, but their use cases have previously been restricted due to their access to outside data.Most blockchains are neither aware of anything going on in the real world , nor can they access any data not native to the chain itself.

The data that they could not previously access includes any data available on the traditional web, as well as those accessible through APIs. When you start to consider just how many of the products and tools we use today rely on these data, the problem becomes apparent.

While there has been a multitude of efforts to solve this issue, most of the solutions have come to meet at least one of three main limitations.

1. Centralization
2. Existing data solutions such as API feeds and some other oracle solutions are centralized by design. This not only goes against the ideology of decentralization and trustlessness but also represents a severe potential security flaw. Relying on a central authority to report data means that you are exposing yourself to the possibility of data manipulation and outages, both of which can have catastrophic implications on any services that depend on it, not to mention on the end users themselves.
3. Network Congestion
4. Most of the existing oracle solution of them are constrained by network congestion. This is mostly the result of the solution being on the same blockchain as the application itself -- competing for block order. Thus, if the blockchain’s network were to become full with pending transactions, the data request transaction themselves would also be delayed.
5. High Cost
6. they are expensive. This comes from both the cost to research, develop, and deploy the solution, as well as the various costs associated with operating and maintaining it in the long run. Previous High-Level Overview Next The BandChain Oracle