

# BetOnChain

Betting on Sport/Esport events

---

Rostik#1223

journal#6461

# Introduction

The sports/esports betting market is valued at more than 75 Billion dollars. Such a market is sometimes considered to be manipulated, not very transparent, etc.

The blockchain, with its unique technology, may allow to add transparency into this market and hence, make it more efficient

Augur (built on Ethereum) faced scalability issues. Substrate can solve such issues

Introduction

Our idea / architecture

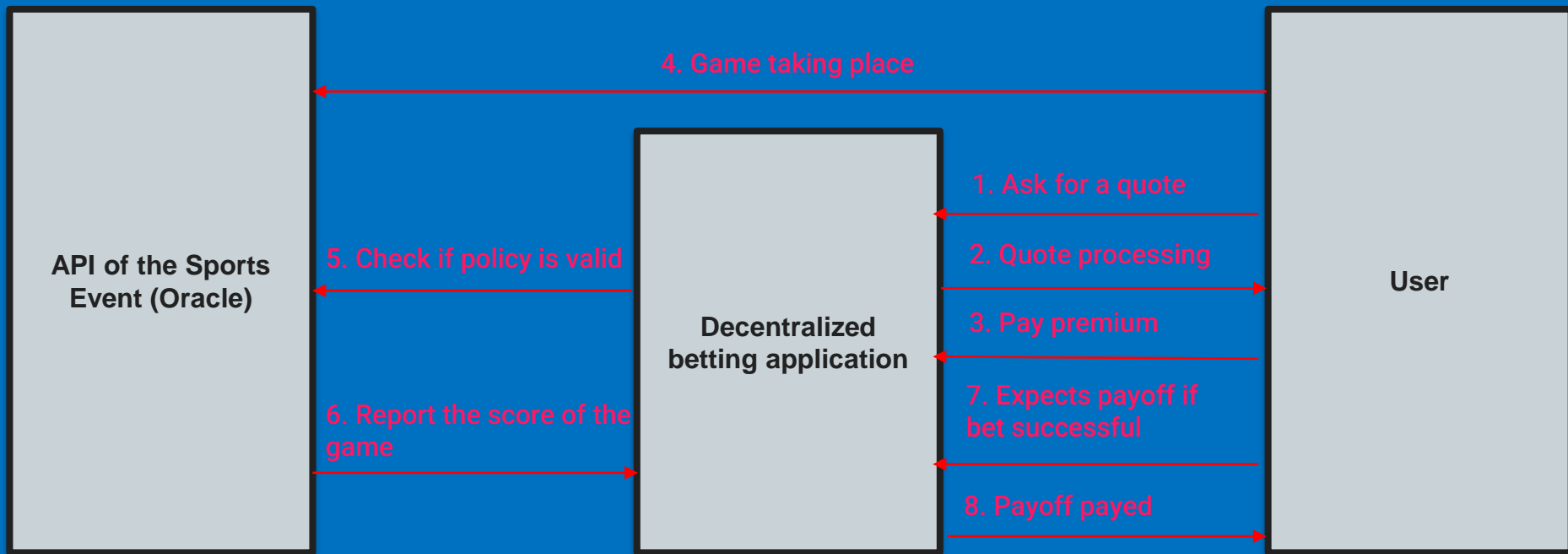
What we have done

Challenges encountered

Next steps

Demo

# Our idea / Architecture



Introduction

Our idea / architecture

What we have done

Challenges encountered

Next steps

Demo

# What we have done

Base logic: Structure to bet and determine what happens when the match is finished

- The pallets **BetonEvents** and **Beton** have been created
- The pallet **treasury** is also used

Work ongoing for the connection with Chainlink

- Chainlink nodes running on postgresql databases
- The set up of the internal initiators, the Aggregated Price Feed External Adapter, as well as bridge of the substrate blockchain with the Chainlink node has been done

Introduction

Our idea / architecture

What we have done

Challenges encountered

Next steps

Demo

# Challenges encountered

## Obsolescence / Depreciation

- The data providers (Chainlink) with Substrate. Documentation is no longer so relevant - only the general structure can be used
- Dependencies: Libraries are depreciated

## Compatibility with macOS

Introduction

Our idea / architecture

What we have done

Challenges encountered

Next steps

Demo

# Next steps

Make sure we can  
connect to Chainlink  
data feed

Optimise the pallets  
and win-rates

Make a proper frontend

Refresh the tutorials - to  
provide a **convenient**  
and **updated** possibility  
for users **to use feeds**  
**from Chainlink with**  
**Substrate**

Introduction

Our idea / architecture

What we have done

Challenges encountered

Next steps

Demo

# Demo