Worlds.db: a Proveably Fair Database

Ryan Walker
ryan.cjw@gmail.com

Abstract. This paper outlines how to move a traditional trusted database into the decentralized world. This yealds the advantege of using existing database structure, ie: SLQ and Mongo - without having to reinvent the world.

1 Introduction

Before I start, I should say there are several centralised components of this y architecture. However I chosen to build it's implementation as a "phase zero" of Worlds.

Blockchain infrastructure is being used to build decentralised games. However most games are build using traditional database infrasture. Building technology has always been about getting to market fast, building your MVP first and the focusing on the core tech once the world realised it's capabilities. Most gaming projects in the space have moved straight to the ideal without answering the core question of: How will you get gamers?

The anwer to the questions is simple, you need to build incrediable games. Execution on the answer is not simple, the gaming market is dominated by huge companies. Unless you have a silver bullet of a game, you're going to be hard pressed to drive adoption.

The quickest way to drive adoption is to take existing MMO infrastrucutre and build around it. This is why I chose to write the document. I will outlinine how to port traditional databases to reach a near trustless entity,

2 Overview

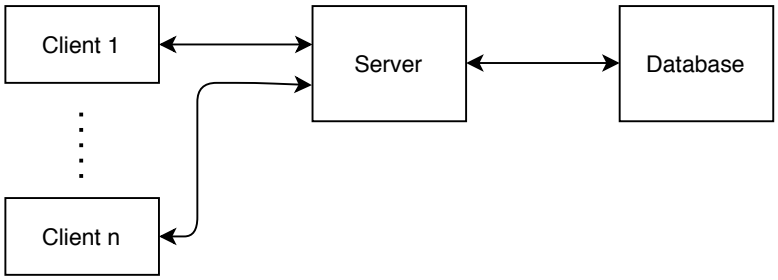
Nearly all MMO games are using a traditional databsese. This is SQL, Mongo or otherwise. These databases rely on a trusted design, meaning whoever hosts the database will not commit to malicious activity. However there is a way to build a ecosystem that relies on a central database without having to trust the contents.

This is called distributed verification, the contents on the datbase are verified on a blockchain, with the heavy lifting done by the database. This also has the added feature of not having to port existing games to having blockchain interaction.

3 Archatecture

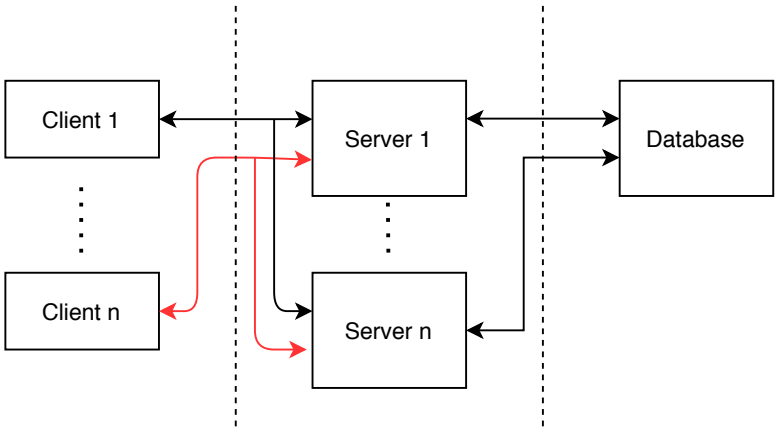
3.1 Existing Design

The existing deign shown in figure whatever is the obvious solutions to a MMO game. The data is kept in a normal trusted database.



3.2 Nieve Design

To crack the problem of a distributed MMO might lead of the nieve design... This will allow player to move from game to game, withe fully different mechanicics.



However there is an obvious flaw. When the number of servers n scales to a high number you have the following issues.

- Servers with malicious intent have write capabilities on the database
- more issues here

4 Worlds.db Archatecture

This is much different
The archatecutre is laid out in figure 1. The clients connect to the