

Polygon Marketplace

Technical Specification

Names:	Dara Lynch Alan McGrath
Project Title:	Polygon Marketplace
Document Type:	Technical Specification
Student ID's:	Dara Lynch - 19324446 Alan McGrath - 19392951
Date of Completion:	03/03/2022

<https://polygon-marketplace-lynchd26.vercel.app/>

Contents

1. Introduction

- 1.1 Overview
- 1.2 Glossary
- 1.3 Proposed Implementation vs. Final Implementation
- 1.4 Testing

2. System Architecture

3. High Level Design

- 3.1 User Diagram
- 3.2 Sidechain Relationship
- 3.3 Context Diagram

4. Issues & Solutions

5. Subsequent Improvements & Developments

6. References

1. Introduction

1.1 Overview

Justification

We chose to create this platform because the cryptocurrency industry is growing rapidly. More and more people are investing and the market capitalization of the industry is near 2 trillion dollars. Many people would like more options in what they can do with their cryptocurrencies. Companies such as Travala have become massively popular allowing users to spend their crypto on booking trips and hotels. Many people wish to be able to spend their crypto on things without having to first convert back to their native Fiat currency, which in itself is a taxable event. Our platform gives holders of crypto currencies almost unlimited potential as to what they can use their holdings to purchase.

Deployment

We ran a local node using hardhat to allow for us to test locally and deploy our contracts. During testing we also deployed to a localhost port allowing for quick updates to the site when the code was changed. Once we were in the final stages of the implementation of the project we deployed to the official Polygon Mumbai Testnet. This means our contracts were publicly deployed and available to anyone using any device. We also deployed the website using Vercel which was extremely simple as we had built using NextJS which was created by those who made Vercel.

Technologies Used

We used the technologies we had previously said in the functional specification that we would use. These were all completely new languages and frameworks to us, we had not used the majority of them except for the likes of HTML and CSS. The web application was built using NextJS which proved to be a relatively straightforward and practical language. The smart contracts were written in Solidity which we also had no previous experience with. We used tutorials to build out the base of our smart contracts and were then able to add to and adapt that in order to fit our platform's functionality.

1.2 Glossary

NFT - Stands for 'Non Fungible Token'. This is a digital asset which is stored on the blockchain and is often used to represent real objects like artwork, music, even digital goods such as exclusive goods in a video game, access to exclusive offers/groups/opportunities and lots more.

Blockchain - A blockchain is a public digital ledger where data is stored.

Ethereum - Ethereum is an open source blockchain which allows transactions through the use of smart contracts.

Polygon - Polygon is an Ethereum sidechain. It essentially bundles up lots of data and transactions before communicating it to the Ethereum blockchain rather than sending every piece of data individually

Matic - The native asset (currency) on Polygon.

USDT - A stablecoin backed by the US Dollar meaning its value is always equal to the US Dollar.

Web 3 - A relatively new technology that utilises machine learning, artificial intelligence and blockchain to improve the internet experience.

Gas Fees - The fee paid to the network to complete a transaction.

TPS - Transactions per second.

EVM - Ethereum Virtual Machine.

Metamask Wallet - This is a commonly used and secure Ethereum soft wallet. It is an extension which you can add to most popular web browsers.

1.3 Proposed Implementation vs. Final Implementation

The final implementation was largely as described in the functional specification. However some changes were made as we learned more about the process of creating the platform and the technologies involved. Users can view sale items, list their own items for sale, view their purchases and sales, message sellers, leave reviews as well as view the reviews that have been received by other sellers. This includes all of the key functionality that was set out to be implemented in the functional specification.

The choice was also made to not use user accounts and rather identify users by their wallet address. The decision was made as this system which we have changed to is the industry standard in similar applications in Web3. It allows users to be completely anonymous, they do not have to provide an email address, name, phone number or any other personal information that could then be susceptible to data leaks and cyber attacks. Even in the case of such an attack on our site, the users' funds are completely safe in their Metamask wallet, which the Polygon Marketplace platform has no access to or to the keys which are used to open a wallet.

In the functional specification it was planned to issue users with an NFT when they reached a landmark number of sales (10, 25 etc.). This feature was not included in the final implementation as it was judged to be unimportant. As the final implementation did not utilise user accounts as explained above, issuing NFTs to profiles would be less effective in proving legitimacy as they may have if there were user accounts.

1.4 Testing

Unit tests were written in order to test our smart contracts. The tests were written using javascript. Solidity coverage was used in order to see how much of the code was covered by the testing. There was a successful test coverage of 92.75% reached.

92.75% Statements 64/69 **57.14%** Branches 8/14 **100%** Functions 11/11 **92.75%** Lines 64

File ▲	
Item.sol	<div></div>
PolygonMarketplace.sol	<div></div>

2. System Architecture

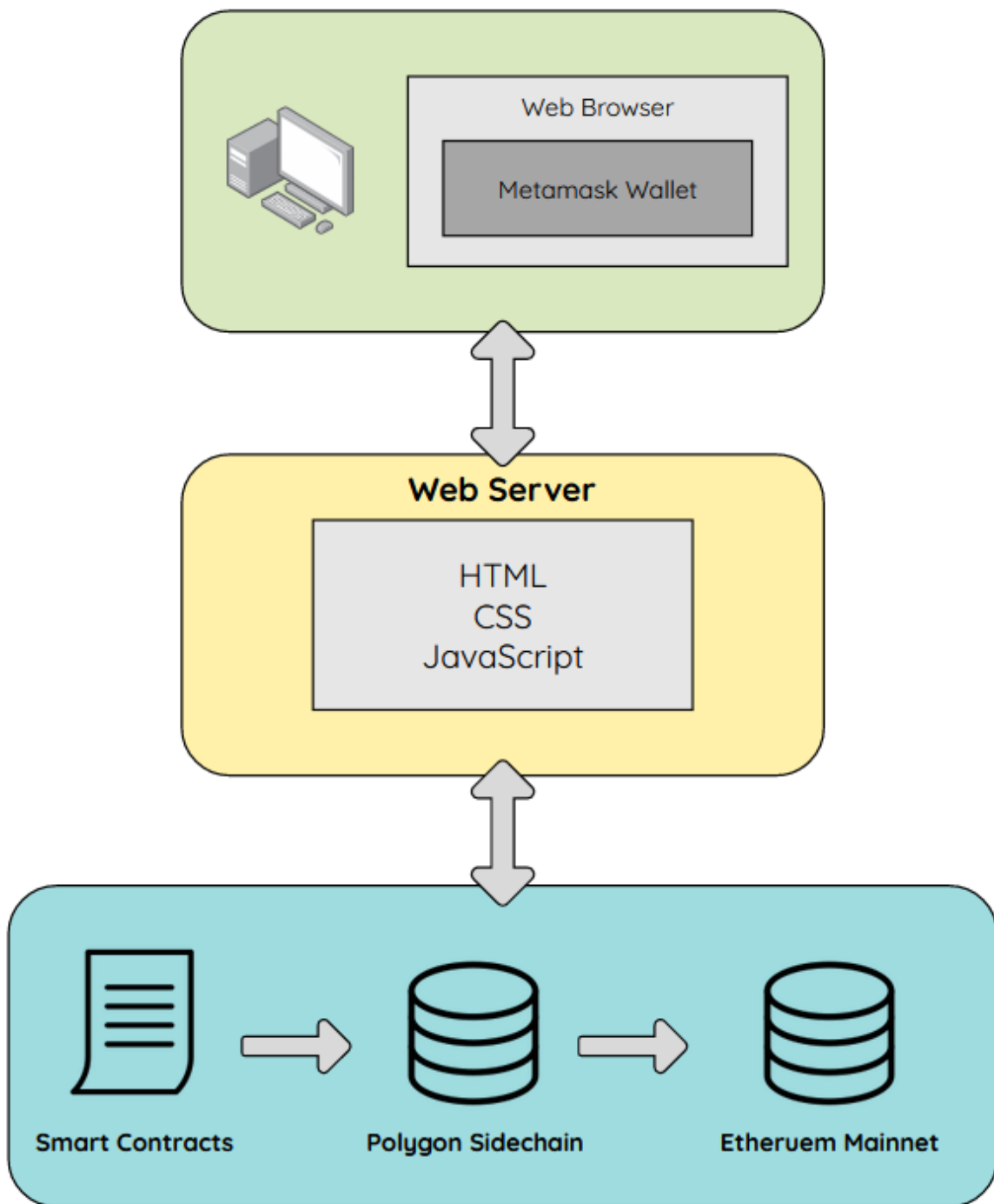


Fig 2.1

Fig 2.1 above shows the system architecture of our platform. A user accesses the website through any web browser which supports the Metamask wallet extension (Chrome, Firefox etc.). The Metamask wallet can contain and transact your crypto assets on the Ethereum blockchain or any Ethereum compatible network. Polygon is an ethereum sidechain which we used ahead of ethereum as it offers faster transactions and lower gas fees.

The browser then displays the HTML and CSS which allows the user to view and interact with the platform.

We have written smart contracts which are essentially a set of instructions to be followed for the transactions on our platform. The transactions take place on the polygon sidechain which is capable of ‘bundling’ transactions and data before communicating it to the Ethereum blockchain, meaning it can fit multiple transactions each time data is sent to Ethereum which allows for cheaper and faster transactions.

3.1 User Diagram

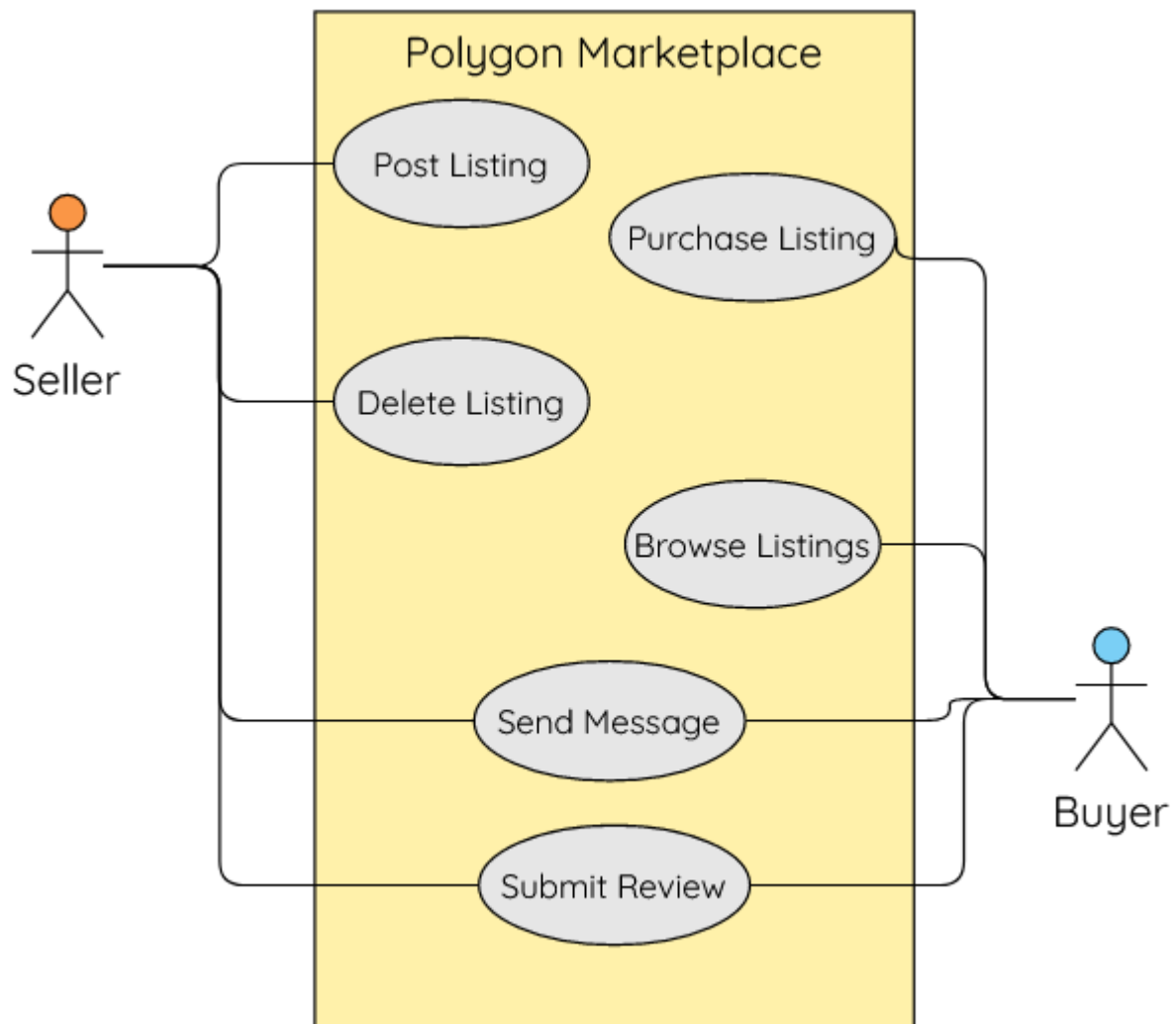


Fig 3.1

The diagram *Fig 3.1* above displays the different actions which can be made by buyers and sellers. Sellers are capable of listing items for sale, receiving messages from buyers and submitting reviews of the buyers. A user acting as a buyer can view and purchase listings, message sellers and submit reviews of their purchases.

3.2 Sidechain Relationship

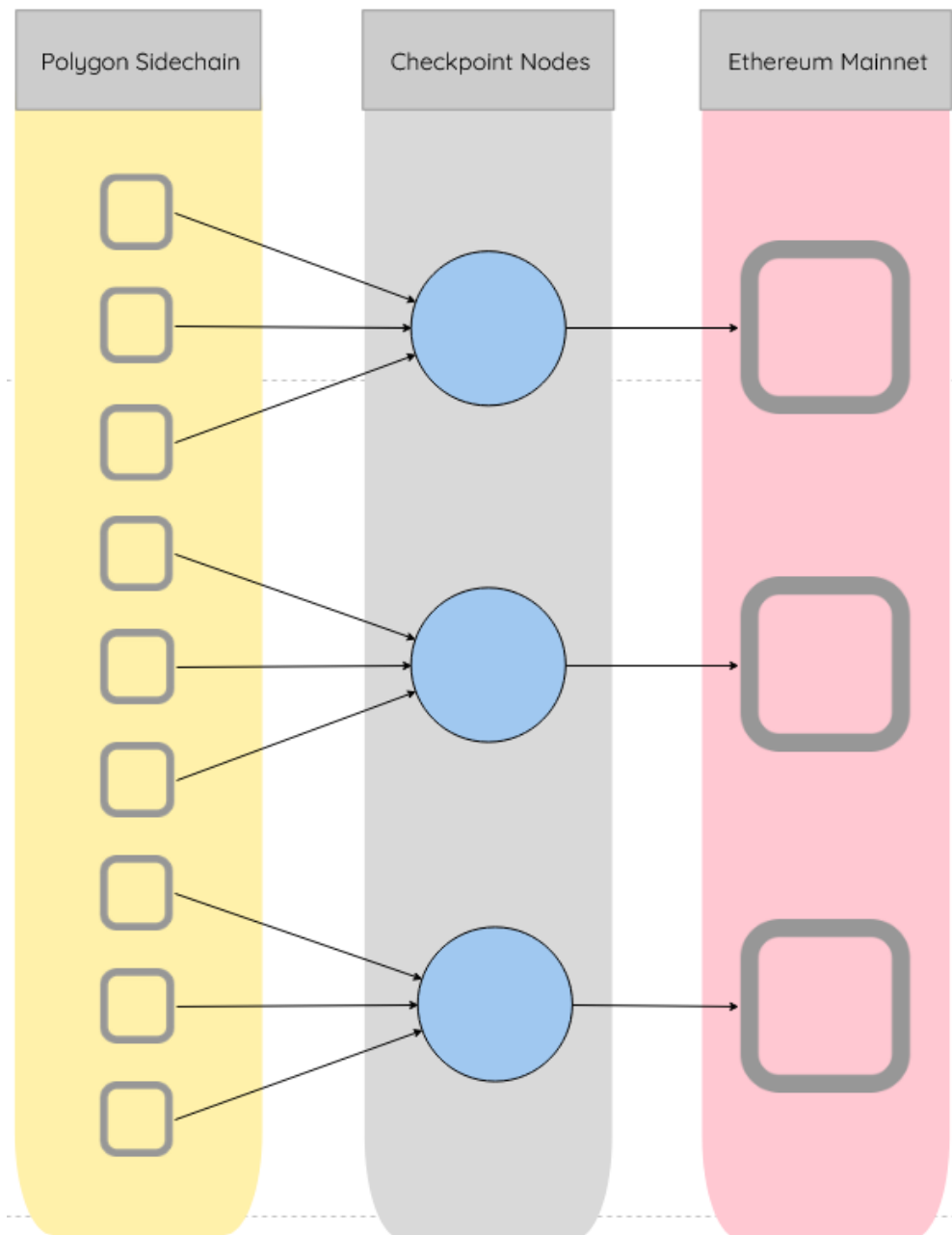


Fig 3.2

Shown in *Fig 3.2*, Polygon is an Ethereum sidechain and its primary purpose is to allow for cheaper and faster transactions on the Ethereum network. It does this by grouping multiple transactions together in each block that is transacted to ethereum. As a result of this it does not need to send data to ethereum after every single individual transaction, and as a result there are far fewer transactions with ethereum necessary. Polygon offers significantly faster and cheaper transactions.

3.3 Process Diagram

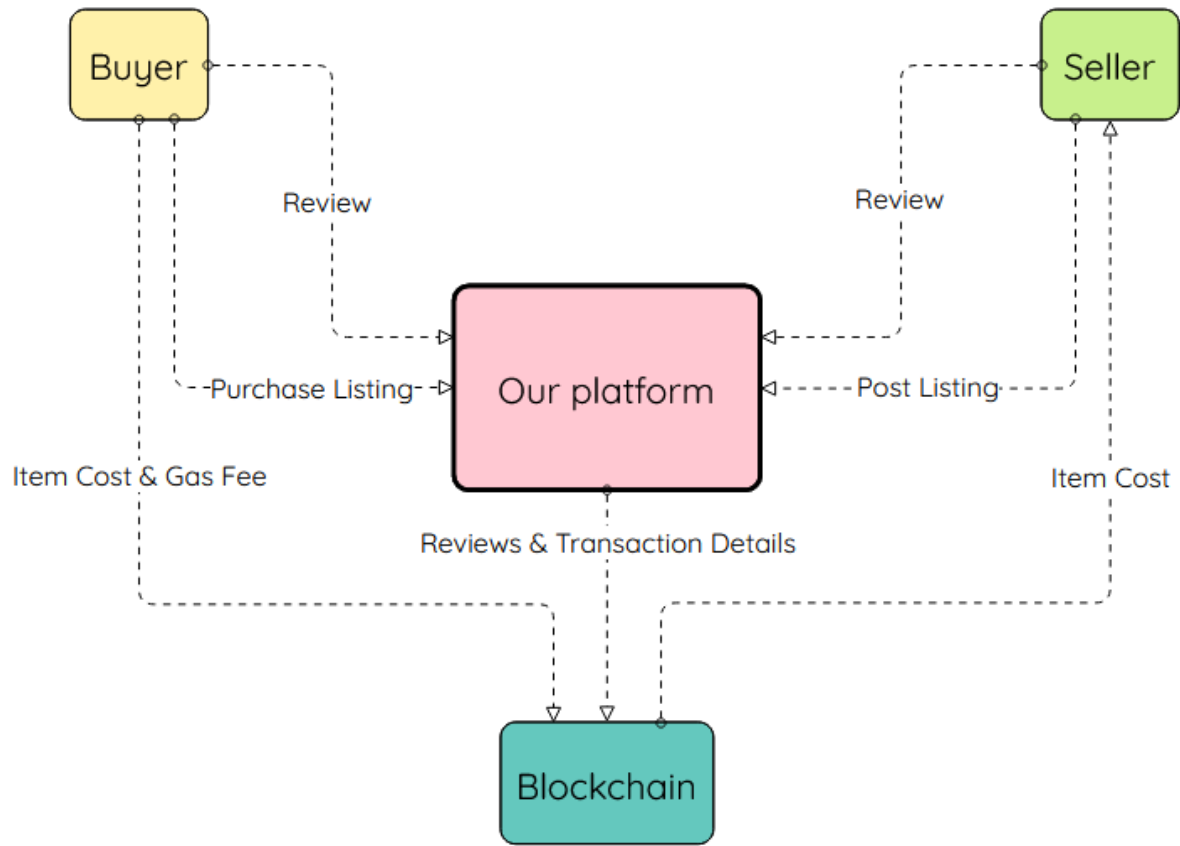


Fig 3.3

Fig 3.3 shows the relationships between our platform, the blockchain and the users of the platform. As displayed, when a buyer purchases an item, the buyer pays the listing price and a gas fee which is transacted through the blockchain and the seller receives the listing price. Both buyers and sellers can post reviews to the platform which are then publicly displayed on our website.

4. Issues & Solutions

The primary issue encountered was in what manner should reviews function. The original plan was that a user would send a review and the person who is receiving the review would get that review token sent to their wallet address. However it became apparent that this would make it more difficult to make the reviews publicly viewable and may also mean someone could transfer the reviews they received which would not make sense functionally. To resolve this issue the decision was made to instead post reviews to the marketplace in a similar way to how sale items are listed. This means the marketplace address owns the review token and hence that cannot be further transacted. This meant reviews could be displayed on the platform in the same way which sale items already had been being displayed.

5. Subsequent Improvements & Developments

There are numerous features which are important to any marketplace like our own which we have not implemented, including deleting listings and sending a seller your delivery address automatically when you purchase an item. Currently both of these functionalities are possible but not in the manner which they ideally should be.

Currently users can essentially delete a listing by purchasing their own listing. This is not a good way to do so but it works nonetheless. In the future functionality would be added to delete your own listings in the proper manner. This would not be very difficult as it would only require a function similar to that which is used when someone buys an item. The function used to buy an item removes it from the marketplace and transfers the price of the listing to the seller, only slightly different from how deleting a listing would work.

Another improvement that could be made in the future would be to improve user experience on mobile and tablets and make the website more adaptive for different screen sizes as we did not manage to optimise this within our timeline.

Finally a profanity filter could be added to the website in the future in order to prevent inappropriate listings or reviews on the platform. Also the possibility of enabling the owners of the platform to delete listings when necessary should be considered. This may not be possible while respecting the decentralised nature and fundamentals of Web 3 platforms.

6. References

<https://docs.polygon.technology/docs/develop/getting-started>

<https://dev.to/dabit3/building-scalable-full-stack-apps-on-ethereum-with-polygon-2cfb>

<https://soliditydeveloper.com/deploy-to-matic>

<https://github.com/sc-forks/solidity-coverage/tree/0.6.x-final#solidity-coverage>