Lab 2 - Art Guardian Product Specification

Carlo Diaz

Old Dominion University

CS411W, Fall 2022

Professor James Brunelle

17 October 2022

Version 1

Table of Contents

List of Figures

List of Tables

1 Introduction

Non-fungible tokens (NFTs) have increased in market value and popularity, especially during Q1 2021. In Q1 2020, the tradeable volume within NFT marketplaces was less than $100 million; this dramatically increased to $1.5 billion in Q1 2021 (Chang, 2021). Throughout 2021, NFTs have consistently increased in popularity and trading volume, eventually causing NFT marketplace volumes to reach $41 billion by the end of 2021 (Collins, 2021).

However, a major problem within the scope of NFTs was the copyright infringement of digital artwork. People mint artwork into NFTs without the original artist's permission and can make a large profit from it. DeviantArt, a popular platform for hosting and sharing digital artwork, reported 30,000 probable counterfeit thefts of digital artwork in November 2021; this number increased to 90,000 probable counterfeit thefts, tripling the number of thefts within a single month (Beckett, 2022). These thefts become a major issue considering the high profitability of NFTs because approximately 50% of all NFT sales were over $200 at the beginning of 2022 (Dailey, 2022).

Currently, there is a lack of security against NFT minting and digital artists do not have an effective solution in protecting their artwork. Digital artists must perform a manual process in order to have NFTs taken down by NFT marketplaces, which can be very tedious. The artist must first have knowledge of their artwork being minted and sold on NFT marketplaces, in which the artist may not even know about these actions being performed on their art. Next, they must manually fill out a Digital Millennium Copyright Act (DMCA) takedown request to the NFT marketplace where the minting or transaction occurred, and must repeat this process for every single marketplace the artwork is found on. Given that there is no effective way to

prevent an artwork from being minted without the original artist's permission, a victim of art theft will most likely find this process to be tedious and frustrating.

1.1 Purpose

Art Guardian is a desktop web application intended to protect the integrity of NFT transactions and protect digital artwork from NFT minting without the original artist's consent. The product prevents the future sale of counterfeit NFTs by providing an automated system that periodically searches through NFT marketplaces and performs image matching to compare the images on Art Guardian's database and the artwork on a given NFT. Art Guardian also auto-fills DMCA takedown requests with provided fields for an intuitive process of issuing takedown requests. The goal of Art Guardian is to grant digital artists an automated process that can halt further NFT profits from being made on stolen artwork.

1.2 Scope

Art Guardian aims to prevent stolen artwork from being sold as an NFT in order to maintain integrity in NFT transactions. The prototype shall do this by performing various algorithms that track art uploaded onto the database and comparing it to artwork found on NFT marketplaces.

Art Guardian's prototype will fully implement art upload, whitelisting, art tracking, image matching, theft alert, DMCA generation, and DMCA cataloging. The prototype will partially implement DMCA filing because these DMCA requests will be sent to test emails instead of real NFT marketplaces. The prototype will eliminate user verification and DMCA tracking.

1.3 Definitions, Acronyms, and Abbreviations

- **NFT**: A sellable, tradeable, non-fungible token that exists on the blockchain and represents some form of data
- **Blockchain**: A decentralized, immutable, public ledger that is split among multiple computers
- **Art Platform**: A site that artists use to publish their art
- **DMCA (Digital Millennium Copyright Act) takedown**: act of taking down a copyrighted work from a website on behalf of the owner of that work
- **Minting**: Using a piece of data, such as an image, to create a unique NFT
- **NFT Marketplace**: website where NFTs are sold
- **AWS (Amazon Web Services)**: Largest provider of various cloud computing services
- **AWS Amplify:** an AWS service for building full-stack web applications
- **Amazon RDS (Relational Database Service)**: cloud-based database service which can work with other AWS services
- **Amazon DynamoDB**: cloud-based database service which can work with other AWS services
- **NFTport API**: interface for working with popular NFT blockchains and markets
- **React**: open-source, front-end JavaScript library for creating websites with modern user interfaces
- **Stripe API**: programming interface for verifying a user's identity
- **OpenCV**: open-source computer vision library for Python
- **Gmail API**: a programming interface for the creation and sending of emails
- **MySQL**: a relational database management system

- **Git**: version control system for tracking software changes
- **GitHub**: online hosting of the git version control system
- **JavaScript**: a general purpose programming language often used for web development
- **HTML**: markup language for displaying documents in the web browser
- **CSS**: style sheet language that specifies the style and layout of how documents are displayed in a web browser

1.4 References

Collins, B. (2021, December 23). Fungible vs Non-Fungible Tokens: What's The Difference? Bryan Collins. https://www.bryancollins.com/fungible-vs-non-fungible-tokens/

BBC. (2021, March 12). What are NFTs and why are some worth millions? BBC News. https://www.bbc.com/news/technology-56371912

Telmo Subira Rodriguez. (2018, December 2). Blockchain for Dummies. Medium; The Startup. https://medium.com/swlh/blockchain-for-dummies-d3daf2170068

Coincorner. (2022, February 11). What Is Nft Minting? Coin-Corner. https://coin-corner.com/what-is-nft-minting/

Beckett, L. (2022, January 29). 'Huge mess of theft and fraud:'artists sound alarm as NFT crime proliferates. The Guardian. Retrieved January 31, 2022, from https://www.theguardian.com/global/2022/jan/29/huge-mess-of-theft-artists-sound-alarm-theft-nfts-proliferates

Dailey, N. (2022, January 6). NFTs ballooned to a $41 billion market in 2021 and are catching up to the total size of the global fine art market. Markets Insider. Retrieved February 23, 2022, from https://markets.businessinsider.com/news/currencies/nft-market-41-billion-nearing-fine-art-market-size-2022-1

Abrol, A. (2022, January 14). What is an NFT marketplace and How Do You Create Your Own? Blockchain Council. Retrieved February 23, 2022, from https://www.blockchain-council.org/nft/nft-marketplace/

Mr. DMCA Helper. (2022, February 23). What is a DMCA Takedown? Dmca.com. https://www.dmca.com/FAQ/What-is-a-DMCA-Takedown?ref=why_is_sol5a32

Palmer, R. (2022). @arvalis. Retrieved 1 March 2022, from https://twitter.com/arvalis/status/1369230566843813891?s=20

Chang, J. (2021, April 12). How Did NFTs Become So Popular? Medium. https://medium.com/geekculture/how-did-nfts-become-so-popular-f894eea22f90

1.5 Overview

This product specification provides the hardware and software configurations, external interfaces, capabilities, and features of the Art Guardian Prototype. The remaining sections of the documents include detailed descriptions of the hardware, software, and external interface architectures of the Art Guardian prototype. It will also include the key features and the parameters that will be used to implement the feature as well as the performance characteristics of that feature in terms of user interaction, display, and output**.**

2 General Description

2.1 Prototype Architecture Description

The Art Guardian prototype will be hosted using AWS Amplify, which shall host both the web application and the backend environment. AWS Amplify is capable of connecting the frontend to the backend, allowing the client to call any APIs needed for any operation. AWS DynamoDB shall be used as the database of the Art Guardian prototype and it will store end users' information, profiles, and artwork. The prototype's backend shall use third-party libraries and dependency APIs in order to assist with the functional features of the prototype, whether fully implemented or partially implemented. The GraphQL APIs shall be used to connect the Art Guardians' database to its web server.

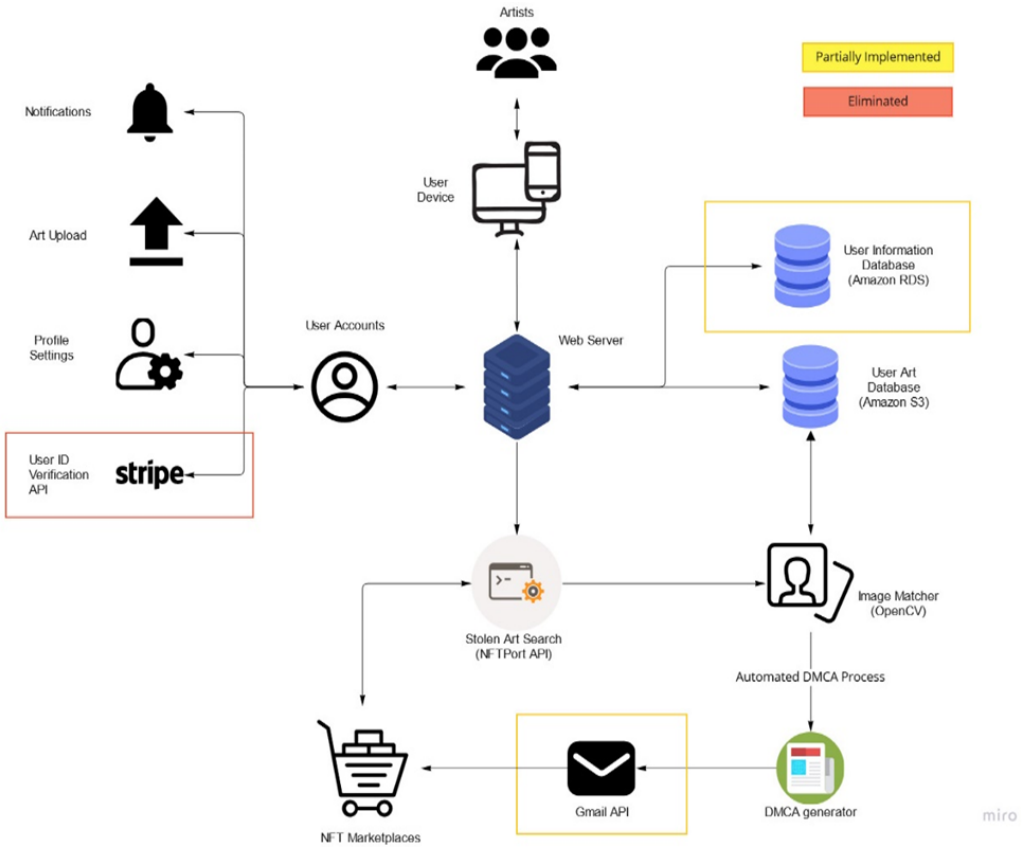

Figure 1. Art Guardian Prototype Architecture Diagram.

2.2 Prototype Functional Description

AWS Amplify shall be the web server that deploys the Art Guardian web application, and it also holds the backend REST APIs that shall be implemented. Users must have an account in order to access the Art Guardian prototype's features. Alerts and push notifications shall be sent to the user for art theft detection and DMCA status updates. Users shall upload images of their art to be stored on the database, where the backend fetches these images and performs image matching and NFT marketplace detection. The database shall also store and encrypt account information. The NFTPort API shall assist with NFT marketplace detection. The OpenCV Python library shall be used during image matching. Once the backend detects that an image on the prototype's database has been minted, it shall initiate the DMCA generator feature, which automatically generates a DMCA takedown request with a fixed format while including the correct information of the NFT. A Gmail API shall be used to send the DMCA request; this feature will only be partially implemented and the DMCA requests will be sent to a testing email.

Features	**Description**	**Prototype Implementation**
Account Controls		
Account Creation	Create account for users to utilize Art Guardian features	Fully Implemented
GUI		
Art Upload	Users upload images of artwork to	Fully

	the database	Implemented
Image Library	Users view the artwork they have uploaded	Fully Implemented
Whitelisting	Remove art previously uploaded to the database	Fully Implemented
Automated Processes		
Marketplace Monitoring	Checks NFT marketplaces periodically for image matching	Fully Implemented
Image Matching	Matches images from NFT marketplaces to images on the database	Fully Implemented
Stolen Art Alert	Send the user notifications of stolen art and DMCA request statuses	Fully Implemented
DMCA Takedown		
DMCA Generation	Generate DMCA takedown notices that are automatically prefilled based on a template.	Fully Implemented
DMCA Filing	Issue generated DMCA takedown notice to the offending NFT marketplace through the Gmail API.	Partially Implemented

DMCA Cataloging	Catalog and monitor all generated DMCA takedown notices.	Fully Implemented

Figure 2. RWP vs Prototype Diagram.

2.3 External Interfaces

Certain external interfaces shall be required for a user to successfully navigate and operate the web application.

2.3.1 Hardware Interfaces

A desktop or laptop running Windows, MacOS, or Linux that has access to the Internet shall be required to utilize the Art Guardian prototype.

2.3.2 Software Interfaces

The prototype shall utilize various software interfaces to assist with full-stack development. AWS Amplify hosts the backend, frontend, and database of the prototype. Github shall be the software collaboration tool for version control. The frontend shall be coded using the React framework and utilize UI components provided by AWS Amplify. AWS DynamoDB shall be used to store users' art and profile information. The backend shall use the NFTPort API and OpenCV library for NFT image matching and art detection. A Gmail API shall be partially implemented for sending the DMCA request to a testing email.

2.3.3 User Interfaces

The prototype's web application shall require a mouse for desktop navigation and a keyboard for data input. Any traditional monitor shall be required.

2.3.4 Communications Protocols and Interfaces

The prototype shall be hosting using the HTTPS protocol for secure communication over the internet and maintaining security during the authentication process. TCP/IP protocols shall be used for communications between the internet and the user's desktop or laptop.

3 Product Requirements

3.1 Functional Requirements

3.1.1 User Interface

3.1.1.1 Account Creation *(O: Diaz, M1: Ihde, M2: Thompson)*

- Users that do not have an account shall create an account through the sign-up page.
- This function shall create and store accounts with a username, password, confirmed email address, and linked art account.
- An account shall be required for users to utilize features of Art Guardian.
- The linked art account needs to originate from either Twitter, Instagram, Tumblr, or DeviantArt.

3.1.1.2 Art Upload *(O: Ihde)*

- This function shall allow users to upload images of their artwork to be stored on the database, which will then be available to view in the art library.
- Users must upload images that are in PNG or JPG formats.

3.1.1.3 Image Library (*O: Ihde, M1: Zheng*)

- This feature shall display all of the images uploaded by the user. Images shall be displayed as cards in a rectangular grid format that adjusts its number of images per row based on the window size. Each card will display a preview of the uploaded art and the

following information will be displayed under the image preview:

- Title
- Upload date

- Additionally, there shall be an edit button which when clicked, a pop up window will appear to the right of the art, and users can edit the title, set whitelisting status, or delete the art piece, which will also remove it from the art database.

3.1.2 Algorithms

3.1.2.1 **Whitelisting** (*O: Roberts, M1: Thompson*)

- This feature allows users to exclude specific NFTs from results based on Token ID.
- If the Token ID is found, the NFT is added to the whitelist.
- If the Token is not found, the application will state that the Token ID is not valid and to insert another ID.

3.1.2.2 **Marketplace Monitoring** (*O: Roberts, M1: Diaz*)

- This feature shall run on the backend to monitor NFT marketplaces for stolen artwork on a minted NFT.
- This feature will utilize the NFTport API to monitor popular marketplaces such as Opensea, rarible, and binance.

3.1.2.3 **Image Matching** (*O: Zheng, M1: Ihde, M1: Thompson*)

- This feature shall run as a AWS Lambda function which connects to the database.

- The feature relies on the NFTPort API for returning images that will be compared to. Each image uploaded by the user shall be searched using the NFTport API, which will return multiple NFTs that best match the uploaded image. Only the top five NFTs will be passed on to the BRISK feature-matching algorithm.
- The BRISK feature-matching algorithm will compare the image uploaded by the user to the top five NFTs returned by NFTPort API. The BRISK feature-matching algorithm will return a similarity score for each comparison, and the NFT with the highest similarity score, along with the user uploaded image, will be passed on to the Functional Requirement 3.1.2.4.

3.1.2.4 **Stolen Art Alert** (*O: Zheng, M1: Thompson*)

- Provide the capability to receive push notifications on desktop environments able to run web applications. The information within the push notifications will include:
    - The image uploaded by the user and the NFT with the highest similarity score, returned by the process described in Functional Requirement 3.1.2.3.
    - The online location, listed price, and uploader identity of the NFT.
- Users can respond to the push notification by

- Confirming and initiating the DMCA Generation and DMCA filing processes mentioned in requirements 3.1.3.1 and 3.1.3.2.
- Whitelisting the image, described in requirement 3.1.2.1.
- Ignore the push notification.

3.1.3 DMCA Takedown Processes

3.1.3.1 DMCA Generation (*O: Roberts, M1: Diaz*)

- This is a page that will automatically generate a DMCA take down for a user after the user confirms the stolen artwork is theirs and that they would like to pursue a take down. The take down will be generated using the preset DMCA Takedown Template as well as data collected from the NFT that was found to be stolen art and user data. This includes:
  - User artwork title
  - Marketplace name
  - Marketplace URL
  - email address of notice sender
  - A statement that the notifier has a good faith belief that the material is not authorized by the intellectual property or copyright owner, its agent, or the law.
  - Statement that Art Guardian has permission to send this notice on behalf of the copyright holder

3.1.3.2 DMCA Filing (*O: Roberts, M1: Diaz*)

- This feature shall send the DMCA takedown request through email. This feature shall be partially implemented in the prototype, in which the DMCA takedown request will be sent to a testing email via the gmail API.

3.1.3.3 DMCA Cataloging (*O: Roberts, M1: Diaz*)

- This function tracks and catalogs information regarding generated DMCA takedown notices.
    - This includes:
    - Date and time of generation and issue
    - Parties involved
    - DMCA sent
- This catalog can later be accessed by the user via the user profile under “DMCA History”

3.2 Performance Requirements (*O: Diaz, M1: Thompson*)

Initial startup should not take more than 30 seconds and each navigation through pages/ sections of the application should not take more than 5 seconds. DMCA generation should not take longer than 30 seconds. Art Guardian should be able to store and track 100 pieces of artwork without skipping scans.

3.2.1 Scanning Frequency (*O: Zheng, M1: Ihde*)

- Art Guardian will scan NFT marketplaces for counterfeit NFTs using artwork uploaded by the user once every two weeks.

3.2.2 Web Application Performance

3.2.2.1 Web Pages (*O: Zheng, M1: Kidd*)

- All web pages will load within five seconds.

3.2.2.2 Image Upload (*O: Zheng, M1: Kidd*)

- Uploading images shall not take more than one minute.

3.2.2.3 Image Loading (*O: Zheng, M1: Kidd*)

- All entries within the Image Library will load within one minute.

3.2.2.4 Image Matcher (*O: Zheng, M1: Thompson*)

- Results will be returned from the Image Matcher within one minute.

3.2.2.5 DMCA Generator (*O: Zheng, M1: Thompson*)

- DMCA takedown notices will be generated within 30 seconds.

3.2.2.6 Network Performance (*O: Zheng, M1: Kidd*)

- Given that the user has a stable internet connection with a speed >700 mbps, all operations involving the transfer of data from a non-UI component to the UI and vice versa will not require more than two minutes in completion time.

3.2.2.7 Concurrency Performance (*O: Ihde, M1: Zheng)*

- The server shall be able to handle at least 5 requests per second.

3.3 Assumptions and Constraints (*O: Kidd, M1: Ihde)*

Condition	Type	Effect on Requirements
Users cannot occupy more than one profile.	Constraint	Bounds the problem of matching users to available profiles

Only valid data entries will be provided.	Assumption	Allows for minimal error checking for the purposes to developing and demonstrating the prototype
The desktop web application will be hosted through AWS services.	Dependency	The AWS Amplify platform must be simulated if AWS cloud services are not available

Table 2. Effects of Assumptions, Dependencies, and Constraints on Requirements.

3.4 Non-functional Requirements

3.4.1 Security (*O: Kidd, M1: Roberts, M2: Ihde)*

- The website will use HTTPS to encrypt transferred data between client and server.
- Art Guardian shall implement Database access control using an MFA requirement and AWS IAM role policies.
- Art Guardian shall secure accounts using two factor authentication via google authenticator.

3.4.2 Maintainability (*O: Kidd, M1: Hite)*

- The Art Guardian uses the Amazon Amplify platform to provide a low-maintenance configuration for accomplishing NFT Uploading and Tracking. The Art Guardian Amplify project is updated on a quarterly basis to provide the very latest updates for NFT monitoring and detection. Maintenance procedures for all other components, such as the database, are conducted semiannually and can be performed by personnel appropriately skilled.

3.4.3 Reliability *(O: Roberts, M1: Diaz)*

- Art Guardian art upload will be available 24 hours a day, 7 days a week. A marketplace search will be done once when art is uploaded then once a month from then on. Art searches will happen in batches of at most 100 pieces of art at off peak hours, 12AM- 6 AM EST. The prototype must be able to complete at least 90% of its searches, DMCA generations, and dispatches without error.

- Art upload and DMCA generation must be available at all times. Tutorials and tips and tricks may go down as long as the aforementioned features are still available without affecting critical performance.
- Art Guardian must have a 90% reliability so theft does not go without notice and artists are able to act. Artists must be able to upload their art and have it tracked and DMCA's generated without interruption as that is the core function of Art Guardian. Tutorials and tips and tricks sections are non critical as they support artists in protecting their art but do not add to the core features of Art Guardian.