

Design Manual

Harry Dunne
C00220865@itcarlow.ie
Supervisor
Patrick Tobin

Abstract

Digital Inheritance refers to the transfer of assets which exist on electronic systems. These assets, making up the owner's digital estate, can include passwords, usernames, online accounts, contracts, receipts, financial transactions, and medical information. The transfer of a digital estate should occur when there is a prolonged, or permanent absence of the original data owner.

Complications arise in digital inheritance because of two reasons – The information being bestowed is confidential in nature and needs to be stored securely, the date at which the information should be transferred is unknown. In many cases, increasing security of stored data decreases future accessibility for beneficiaries, while increasing the future accessibility harms overall security of the data.

Currently, individuals looking for a digital inheritance solution, need to either intrust their data with 3rd party cloud services, or store their information physically so that beneficiaries can find it in the case of an unforeseen event. Both solutions create single points of compromise, that can be exploited by criminals, digitally and physically.

This project aims to address this problem by providing a decentralised solution to the storage and transfer of digital assets. The solution provided has been designed to give users full autonomy of their data and protect them against both digital and physical attacks. Using this technology, an individual can create a highly secure digital inheritance solution within minutes.

Contents

Abstract 1

Section 1 – Introduction 3

Section 2 - Technologies 4

 Application 4

 JAMstack 4

 Components 4

 Section 2.1.1 – HTML 4

 Section 2.1.2 – CSS 4

 Section 2.1.3 – JavaScript 5

 Section 2.1.4 – jQuery 5

 Section 2.1.5 – Bootstrap 4 5

 Section 2.1.6 – Netlify 5

 Section 2.2 – Hyperledger Fabric 5

Bibliography 7

Section 1 – Introduction

This document explores the design elements of building the digital inheritance scheme. The document defines the technologies to be used in the building of the application and its core assets. This document adds onto the UML use cases discussed in the Functional Specification.

Application type - JAMstack

Components:

- **Website:** HTML, CSS, JavaScript, jQuery, Bootstrap 4, Netlify
- **Database:** FaunaDB

This system aims to be decentralised, for this reason, a database is not needed for the inheritance system to work. However, to allow for user signups and the option to backup decryption pieces used in the inheritance system, a database can be added to the website.

Section 2 - Technologies

Application

JAMstack

The JAMstack refers to websites using a – Javascript, API, and Markup technology stack. This technology stack allows for the creation of fully functional websites without depending on a web server, or backend code. Instead it uses Content Delivery Networks, to distribute the site to visitors and decentralised API driven databases, such as FaunaDB to store user data.

This technology stack can be used to create websites with better performance, higher security, easier scalability, and lower price than traditional web development methods.

JAMstack website are also highly portable and open source by design. [1]

Components

Section 2.1.1 – HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web. [2]

Section 2.1.2 – CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is one of the core languages of the open Web and is standardized across Web browsers according to the W3C specification. Developed in levels, CSS1 is now obsolete, CSS2.1 is a recommendation, and CSS3, now split into smaller modules, is progressing on the standardization track. [3]

Section 2.1.3 – JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. [4]

Section 2.1.4 – jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. It is free, open-source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites. [5, 6]

Section 2.1.5 – Bootstrap 4

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. It contains Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery. [7]

Section 2.1.6 – Netlify

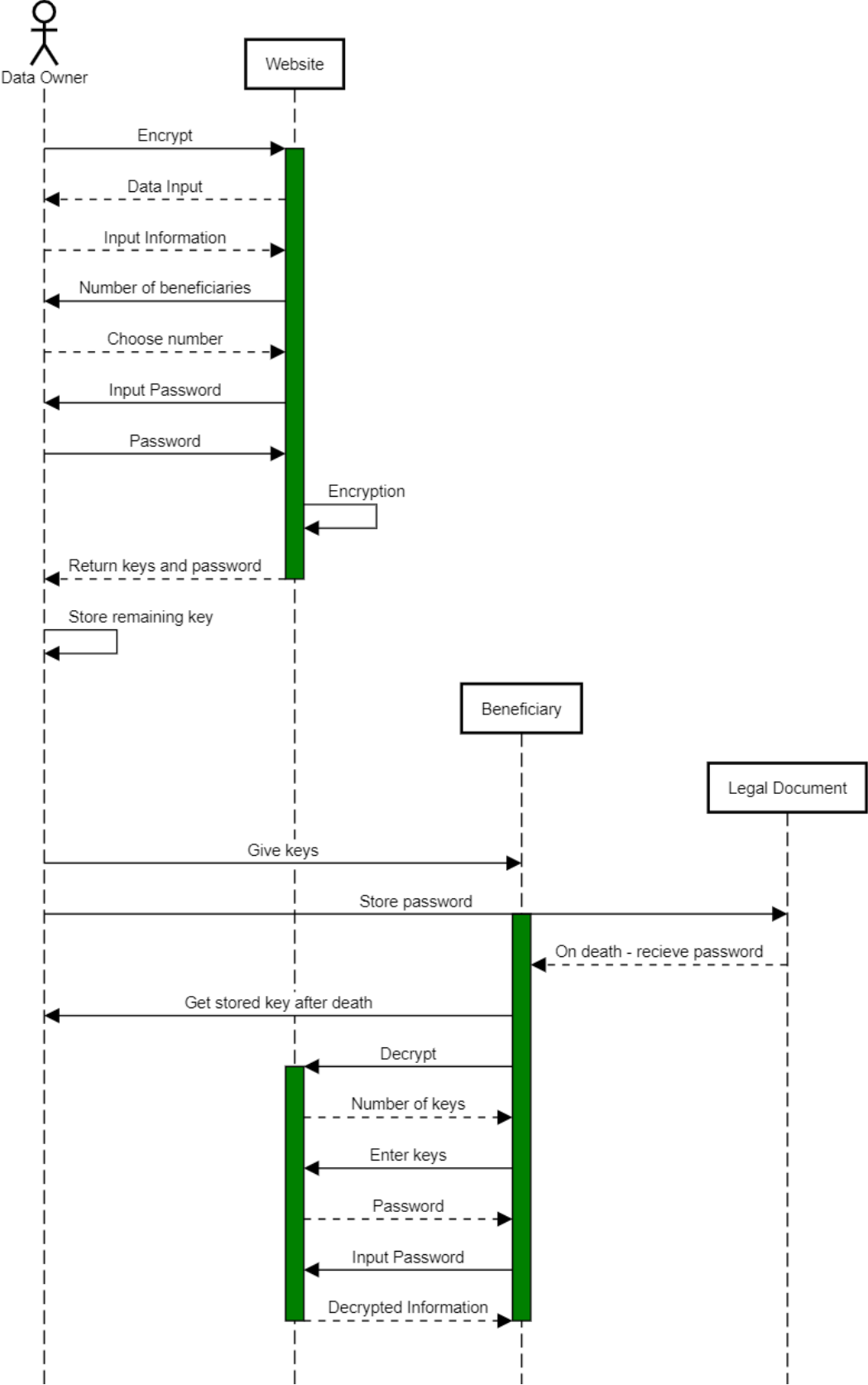
Netlify is a San Francisco-based cloud computing company that offers hosting and serverless backend services for web applications and static websites. It has its own CDN which delivers its users sites to visitors across the world. [8]

Section 2.2 – FaunaDB

FaunaDB is a global serverless database that gives ubiquitous, low latency access to app data, without sacrificing data correctness and scale. It uses GraphQL as its querying language. [9]

Sequence Diagram

Project



Bibliography

- [1] Jamstack.org. 2020. Jamstack | Javascript, Apis, And Markup. [online] Available at: <https://jamstack.org> [Accessed 11 April 2020].
- [2] MDN Web Docs. (2019). HTML: Hypertext Markup Language. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/HTML> [Accessed 28 Nov. 2019].
- [3] MDN Web Docs. (2019). CSS: Cascading Style Sheets. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/CSS> [Accessed 28 Nov. 2019].
- [4] MDN Web Docs. (2019). JavaScript. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Accessed 28 Nov. 2019].
- [5] JS Foundation - js.foundation (2019). jQuery. [online] JQuery.com. Available at: <https://jquery.com/>.
- [6] BuiltWith (2019). BuiltWith Technology Lookup. [online] Builtwith.com. Available at: <https://builtwith.com/>.
- [7] Bootstrap (2000). Bootstrap. [online] Getbootstrap.com. Available at: <https://getbootstrap.com/>.
- [8] Netlify: All-in-one platform for automating modern web projects (2019). Netlify: All-in-one platform for automating modern web projects. [online] Netlify. Available at: <https://www.netlify.com/>.
- [9] Fauna. (n.d.). The database built for serverless. [online] Available at: <https://fauna.com/> [Accessed 11 Apr. 2020].