Mohammed Salman

The CEDD company

Design of national fraud database

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Information Technology

Bachelor's Thesis

13 April 2021

This is the thesis “The CEDD company”. It is an attempt by the author to leverage his day job as a software developer in financial technology, his studies as a software engineer and his personal dream to create his own business. The thesis has been written to meet the graduation requirements for a Bachelor of Engineering degree. The studies at Metropolia University of Applied Sciences were practical in nature and product oriented, and this thesis shares the same nature.

Many people supported and assisted me throughout writing this thesis; am forever grateful for all their energy.

From those who supported me, I want to specifically extend gratitude for a group that supported me the most. Mr. Janne Salonen is a great supervisor that taught me self-reliance and independence, I am thankful for those lessons and much more. Gratitude extends to Mrs. Ulla Patola who supervised the language of this thesis transforming it to human readable text. I also want to thank my colleagues in Holvi Payment services for their patience with answering my questions and explaining of intricate legal matters. Thanks also to my two mentors Tuomas Toivonen and Joonas Peltola who pushed me to finish this thesis.

Final gratitude goes to my life’s beacon and my north star, my family, starting with my parents who spent a tremendous amount of energy raising me, and extending to my brother and best friend Bakr. And finally, I would like to express my gratitude to my dear wife, who is the biggest reason of my maturation. I am certain I would not be writing these words without her support.

Mohammed Salman

Helsinki

22 November 2020

Abstract

Author:	Mohammed Salman
Title:	The CEDD company: Design on national fraud database
Number of Pages:	40 pages + 2 appendices
Date:	13 April 2021
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Supervisors:	Janne Salonen, Head of School

Digitalisation has touched most aspects of the modern human life. The financial sector is one of the most effected by digitalisation with ever growing percentage of commercial transactions conducted online. This digitalisation exposes financial services to increasing fraud attacks. Meanwhile, customer data protection and privacy are other aspects that the financial sector must deal with. The goal of this study is to research the feasibility of leveraging crowdsourcing to implement a fraud database, helping banks with early and ongoing fraud detection and mitigation, hence reducing unrecoverable fraud losses.

The data collected for this study came from multiple sources including legal books, Finnish laws, and financial regulations. The data also included several interviews with professionals from the financial sector. Since the viability of the fraud database as a profitable company is one of the targets, investors were also consulted in this project.

The feasibility of a fraud database is a composite of two criteria, legal feasibility and technical feasibility. During the research of this project, it was found that the project is technically reasonable and that implementing a database for fraud cases is a straightforward task. The legal viability, in contrast, is hard to achieve with the financial sector being a heavily regulated field. It was found that a fraud database provider would need to be a licensed bank, but a bank license / but bank licensing is legally complicated and out of the scope of this project.

The study showed that implementing a national fraud database is possible as a product within an anti-money laundering tools suite. While the fraud database has minimal technical requirements, the legal requirements will increase the operational costs. The study also showed that there is customer interest in a fraud database, with possible service fees covering for the legal operational costs of the database.

Keywords: anti money laundering, fraud, database, financial, digitalisation, crowdsourcing.

`

Contents

`

`

List of Abbreviations

DBMS: Database management system. Software for maintaining, querying and updating data and metadata in a database.

ORM: Object-relational mapping. The set of rules for mapping objects in a programming language to records in a relational database, and vice versa.

Fin-FSA: Finnish Financial Supervisory Authority

CDD: Customer Due Diligence

FinTech: Financial Technology

MVP: Minimum Viable Product

EU: European Union

GDPR: General Data Protection Regulation

AML act: Act on Detecting and Preventing Money Laundering and Terrorist Financing (Finnish Parlement, 2013), Unofficial translation.

GraphQL: Graph query language.

`

1 Introduction

Statistics indicate that half of the world population will transact money online in the year 2021 representing growth of 10% year over year. This growth in the number of users is closely coupled with growth in transacted funds, mounting to 6,685 billion US dollars. (statista, 2021.) his continuing increase in online transaction volume is fueled by companies optimizing their online presence. Online businesses are competing to offer easier user experience. Easier user experience breaks into faster onboarding and check out processes. Equally to improve online business offering, there are other contributing factors converting more offline commerce into online only trade.

Payment card fraud in Europe added up to €1.8 billion in 2016 (European Central Bank, 2018, p. 1). Card fraud is only a fraction of fraud losses reported every year. Fraud attacks are only improving in sophistication and complexity with time. The more transactions are commenced online, the more attack surface there is for criminals to exploit.

Criminals need to extract their illegitimate gains out of the financial system. The extraction of money is known as money laundering, a process for masquerading the source of funds through complicated money transfer operations. Smart criminals are leveraging the complicated online ecosystems for their own schemes, due to the ease of opening bank accounts and naïve legitimate customers. Meanwhile, financial institutions incur most of the losses from fraud attacks, leaving online service providers struggling between complicated regulatory requirements and forever growing fraud losses.

A concept such as crowd sourcing is a familiar concept online yet severely underutilized in the regulated financial industry. By analysing the money laundering process, an easy technical solution was found to counter-fraud, meshing multiple technological concepts to overcome legal and technological difficulties.

`

Currently, banks communicate internally to exchange data on fraud by phone calls. This could be optimized by a database that is accessible by regulated entities to automate the detection of fraud cases. Once a criminal is detected in one bank, other banks are protected. This is a great countermeasure that directs a specific phase of money laundering process called “integration”.

It seems viable and economically profitable for all legitimate parties to use such a database. An entity that manages the database, then, can charge for this extra layer of communication. Banks are increasing their investment in fraud detection and prevention technology. In a survey done by KPMG international, half of the respondent banks noted that 75% of fraud losses are non-recoverable (KPMG international, 2019, p. 5.).

1.1 Business Context

Fraud losses are usually part of the cost of business for financial institutions. There is an arm race between fraudsters and money launderers from one side and financial institutions on the other side. It is always beneficial for service providers to reduce fraud losses. In different markets, different solutions were developed to counter fraud and money laundry. A variety of fraud solutions is caused by the varying legal framework as well as the peculiarity of each market. The aim of this thesis is to come up with a product that would fit the Finnish market as a starting point with an aim to expand in the future.

Finland was chosen as the starting point for multiple reasons. The first was the familiarity of the Finnish financial sector for the author. The obvious lack of competitors was the second contributing factor. The third factor was the transparency of the clear communication channels of the Finnish state as well as translated laws and regulations.

The financial sector is incurring large losses over money laundry and fraud (KPMG international, 2019, p5). While the losses are written off as operational cost, they open a space which can be filled by a database product. A similar

`

fraud database product has been implemented in the British market with great success since 1988 (CIFAS, about us). The idea is simple, a criminal can have only one successful detected crime before they are restricted in all other financial institutions. Unlike the British CIFAS, there is currently no official means for banks to communicate and share data regarding fraud investigation in Finland.

1.2 Business Challenge

The main challenge facing financial services regarding fraud is the imposed information isolation. Financial information is highly confidential by law. Any information related to any person using a financial service is confidential aside from the legal statue of the person. A criminal is protected by the same confidentiality laws that protect legitimate customers. This confidentiality is abused by criminals that can rotate their operations across different financial service providers to extract as much funds as possible before the eventual slow detection (Sahlin, 2009, p7, p9).

Implementing a database is technically straightforward. Cloud providers such as Amazon web services or Google Cloud provide simple one-click database instances. The complication arises from the nature of stored data. The data of banking customers are legally confidential and should not be disclosed unless for legal matters (Federal deposit insurance corporation) . The nature of the data imposes unique difficulties that must be overcome.

1.3 Thesis Outline

This thesis combines two domains. The first domain is legal, coming from the regulated financial industry. The second domain is technical, concerned with creating a technically feasible product. The first domain requires discovery of fraud together with money laundry. Technical aspects such as database design and interface design are analysed in chapter 2.

`

Analysing the current state starts in chapter 3, dealing with the contemporary situation. The analysis focuses on Finland as the case market. Market competitors such as Suomen Asiakastieto are further analysed. The examination continues with similar markets and their peculiar solutions.

The available knowledge chapter investigates the legal domain. Fraud crimes usually end up with money laundry to masquerade the source of the stolen money. Subsequently, fraud is defined as well as fraud consequences. Money laundry is conducted in three phases, where the second phase is layering. The aim of this project is to prevent the layering phase. Hence, money laundering is investigated with its three phases.

After identifying the legal limitation as well as the operational processes, the study continues to the practical part. First, the database operational process is defined as well as related entities. Based on the suggested process, the thesis describes how the database schema was selected and how other design choices were made. Then, a series of different technologies are discussed along with their implementation details. Multiple architectures, which can be developed into a full product, are discussed. The thesis closes with the evaluation of the proposal and a summary and conclusions.

`

2 Method and Material

This research was made with the help of experts working in the Financial Technology (FinTech) field. This section will outline the sources of data collected for the research and the methods of collection.

2.1 Research Design

This project was carried out in three stages. The first stage was the requirements collection stage. Having a database of private financial information is a task with the high legal requirements. The first stage is to start drifting requirements proposals and iterate the proposals with Finnish Financial Supervisory Authority (Fin-FSA). The first stage outcome is having a legally validated set of requirements. The first set of requirements is the general data protection regulation(GDPR). Since the product will contain data for users within the EU. Although most companies in the EU are under GDPR, the financial sector has even higher legislation to follow. Some of the high legislation to follow include Anti Money Laundering directive (AML directive) (Jan Putnis, 2021.) The section will discuss the bank-customer confidentiality as well.

The second stage is the design and implementation. The aim of the second part is to have a minimum viable product (MVP) to demonstrate to customers. The product can store information about the banking service customer. The service provider can mark a customer with a fraud case when needed. After marking a person for a fraud, any new registration for the fraudulent person will result in a notification for the service provider.

The third stage includes pitching the product and iterating with customers. The customer pitch will be presented as a startup pitch. The third conclusion stage is achieving customer approval. The aim behind this project is having a functional product that can be sold and the customer’s approval is essential.

`

2.2 Project Plan

The lack of proper communication layer for anti money laundry in Finland was noticeable for banking professionals. The project needs to answer two questions. The first question concerning legal feasibility. The second was about implementation details.

The first part requires a large amount of external knowledge. Mainly, the legal domain of data privacy is complex enough to sustain several legal firms in Helsinki alone. Thus, consultation of multiple sources was required. The sources were used to guide the work providing legal requirements for the operation of the database. The first phase also included pitching the products to possible clients. The main goal was to measure market fitness.

The second part is the technical implementation. While this product is in the legal domain, the software solution is built by engineers. Hence, software architecture choices are discussed. The research considers multiple implementation designs. The technical context of the application influences the design choices.

`

2.3 Data Collection and Analysis

The data was collected from meetings and discussions with authority bodies and banking experts. The interviews and communication are shown in the Table 1 below.

Table 1. Details of interviews and discussions. Based on Aittola (2015).

	Participants / role	Data type	Topic, description	Date, length	Documented as
	Data				
1	Respondent 1: Mr.Jari Hautaranta	Lunch meeting	The possibility of National fraud database in Finland. The legal limits and customer susceptibility to such a product.	Dec 26 2018	Field notes
2	Respondent 2: FinFsa Innovation desk	Emails	The legal standpoint for an international fraud database.	Feb 17 2019	Emails
3	Respondent 3: Masnad Neith	Telephone interview	An interview about the current situation based on the respondent's experiences as a senior developer for a FinTech company (aurora exchange).	March 19 2019	Field notes
4	Respondent 4: Piia Sillanpää	Lunch meeting	Finnish Fintech market need for a/the fraud database and the inability of the public sector to fulfill the need.	March 2019, 60min	Recording
5	Respondent 5: Moaffak Ahmed	Real life meeting	Startup pitch for the fraud database.	April 2019 20min	Field notes
6	Moaffak Ahmed	Emails	Legal viability of the national fraud database	April 2019	Email

Several meetings were conducted gathering data for this project. The most relevant meetings are documented in Table 1.

`

3 Current State Analysis

This section presents the current anti-fraud products available for banking sector. The analysis starts with the Finnish market and expand to related European markets.

3.1 Overview of Current State Analysis

The national fraud database is a solution for a problem that the author got exposed to working for FinTech company. Fraud is rampant for banks with online services while anti-fraud efforts feel like an endless war for banking sector. The national fraud database would allow companies in the financial sector to communicate in a timely fashion preventing most financial fraud losses.

The first step was to ensure that a solution is missing from the market. The work started with conducting a series of interviews with banking sector professionals. In these interviews, details of the fraud problems were discussed. The discussion also grew to other subjects like market feasibility as well as legal requirements. During these meetings, the possibility of a fraud database was also explored with the interviewees.

3.2 Finnish National Fraud Database Background

Fraudsters often transfer scammed funds between multiple banks and payment services. The funds are transferred around to masquerade the source of fund. Investigations regarding financial crime are conducted in Finland by the FIU (Financial Intelligence Unit). While each bank report fraud cases separately, there is no automated way for the authority to consolidate all the cases related to one fraud case. The fragmentation of fraud cases is getting even worse with financial fraud cases on the rise. During the last decade, fraud and embezzlement in Finland increased by more than 50% while payment fraud rose almost 40% for the total number of annual cases (Tilastokeskus, 2020). Market specialists shared that the FIU process less than 20% of the reports that banks submit annually. The low process rate causes some cases not to be investigated at all.

`

Each bank has their own format of reporting and investigation. Meanwhile, big banks are restricted from implementing certain type of tools and utilities. The restrictions are enforced by the Finnish Competition and Consumer Authority (FCCA). The aim of these limits is to prevent forming cartels by the banks through a solution that can be considered as non-competitive. In a case from 2016, FCCA rejected terms and conditions (T&C) license that was implemented by several banks. The case cited the FCCA judgment of the license being anti-competing (Finnish Competetion and Consumer authority, 2019), showing that any corporation might be subject to authority scrutiny.

The mixture of stringent competition laws preventing banks from cooperation as well as lack of governmental tooling left a space to be filled in the market for a fraud prevention solution.

3.3 Fraud Databases in Other Markets

Finland is a small market compared to bigger countries with a capitalistic nature. For comparison, the GDP of the UK is ten times bigger that of Finland. As such, the UK has more commerce and a more attractive market for fraud. This bigger market caused the UK to face fraud challenges earlier than Finland. Starting from 1988, an organization was founded to fight fraud. The name of the new anti-fraud organization was CIFAS. CIFAS was built with a simple premise, which was that fraud prevention is not a competitive issue. Fraud prevention should not be the responsibility of the police alone; instead, fraud prevention is a common responsibility.

CIFAS is a not-for-profit organization that operates closely with the government. The organization manages multiple databases related to fraud and fraud prevention. CIFAS works as a proxy between companies that detect fraud and personal customers who might be affected by fraud. CIFAS is a good indicator that a fraud database can work within the EU zone regulations.

`

3.4 Summary of Key Findings from the Current State Analysis

Implementing a national fraud database is possible and feasible. Other fraud databases exist and thrive in other markets. In Finland, there are regulatory hurdles to overcome. Nonetheless, the need for a fraud database is clear and felt by industry professionals.

Regulatory obstacles were unified by the GDPR which regulates most of the data life cycle in the European Union.

The focus of this study is the possible implementation of a fraud database. This includes the legal framework, a technical minimum viable product, and business validation. This solution would be both financially viable as well a positive solution for the fintech sector and the society overall.

`

4 Available Knowledge and Best Practice on Fraud

Fraud is an umbrella term for several crimes. This section starts with fraud definition and diving into types of fraud. The section then discusses money laundering as well since the latter is an essential part of a successful fraud. Furthermore, the section delves into authority and fraud discussing logistical limitations the authorities face when dealing with fraud. The section closes by discussing the wide range of fraud victims and effects on society. Meanwhile, the banking secrecy rules are also introduced.

4.1 What Is Fraud

Fraud is identified as the deliberate act of attaining financial gains by means of cunning and scheming exploited by one or multiple culprits (Gee, 2014 p. 1; The Law Dictionary).

A crime then requires a set of features to be considered as fraud:

- The act must be dishonest in nature.
- The act must result in direct gains for the perpetuators.
- The act must be intentional.

An error or misjudgement that causes damages cannot be identified as a fraud since it lacks the intention. Meanwhile, some intentional damages incurring to a company can fall into a grey zone between fraud and abuse. An employee that is intentionally underperforming reaps personal gain on the dime of the employer. The employee behavior satisfies all the conditions of a fraud. However, damage might be considered small to be fraud. In such small cases, the act is considered “abuse” (Gee, 2014).

`

The central bank of Malaysia summarizes the types of fraud with five types:

- Unlawful ventures over the internet.
- Trading currencies online in black markets.
- Unauthorized withdrawals.
- Illegal use of payment cards.
- Misuse of officials' names.

(Central Bank Of Malysia, 2010)

After collecting illegitimate funds, the fraudsters need to masquerade the source of the fund. The process of masquerading an illegal money source with a legitimate source is called Money Laundering (Gee, 2014). Although Money Laundering is fought strongly by governments by means of Anti Money Laundering laws, fraudsters are still able to extract small sums of money. Most governments have set standards on which financial institutions are required to report suspicious behavior (EBA European Banking Authority, 2018, p. 141). The Finnish legislator has set the reporting limit for transactions with 15,000 euros or more in one or multiple transactions (Finnish Parlement, 2013).

4.2 Authority and Fraud

The lower reporting limit varies from jurisdiction to another. The legal sum which should be reported in Canada is 10,000 Canadian dollars or more (FINTRAC: Financial Transactions and Reports Analysis Centre of Canada, 2017). Meanwhile, the American congress has set the reporting limit to 10,000 US dollars (The United States Congress, 1970). While those limits are not precisely for fraud cases, the limits highlight the scale on which the financial sector reports. For lower sums of money, there is an area for fraudsters to operate freely.

The reporting limits are set because of the resources required to handle the financial cases by the authority. Some authorities have even raised the lower

`

limit to be several million dollars. Each report requires subsequent investigation by the authority. Consequently, the authority resources are depleted (Samociuk, et al., 2010 p. 174). Fraud cases investigation takes months or years depending on the complexity of the case. The complexity of fraud cases is caused because of multiple reasons including the use of recent technologies by fraudsters as well as multi-jurisdiction cases (Cifas, 2018).

For the financial institutions, reporting fraud to authorities is a costly option. Police investigations might take a long time to process. The police might also focus on one aspect of the case that matters the most for the police. Furthermore, providing constant support for the police from the financial service provider side might be time and resource consuming. Moreover, any information related to the fraud process must be disclosed with the police. Therefore, the information disclosure might be cost the financial institution more loss than the fraud case itself. (Samociuk, et al., 2010 p. 28).

4.3 Fraud Victims

Fraud is causing damages for the customers and the businesses as well as the financial service providers. Most customers online trust that their information security is a top priority for businesses. On the other hand, two thirds of businesses reported a raising level of fraud losses (Experian, 2018). Two thirds of the identity impersonation fraud are male victims (see Figure 1) and one third of account takeover victims are over 60 years old (Cifas, 2018). Anyone can become a fraud victim, consequently, early detection of fraud is not only a money issue but a social issue helping real humans with real lives.

`

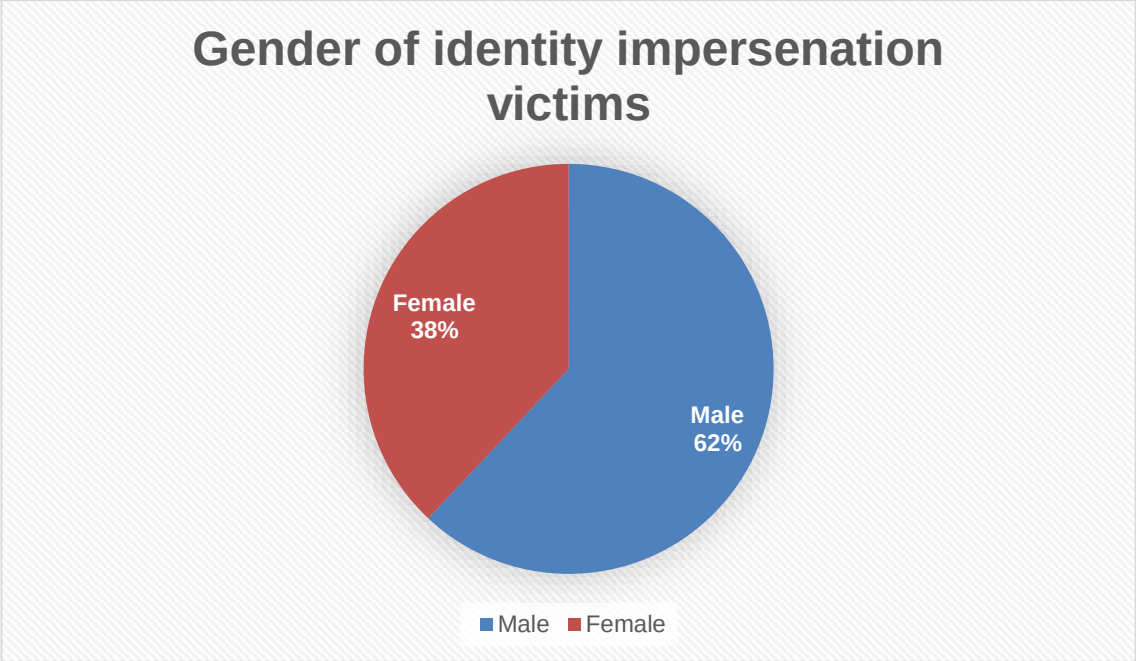

Figure 1. Gender ratio in identity impersonation cases. Adapted from Experian (2018).

On the other side, businesses are being affected negatively by fraud as well. In 2018, 65% of businesses reported detecting the same amount of fraud now or slightly more than before. Furthermore, when asked about their ability to detect fraud, high percentage of companies expressed mistrust in their own fraud detection process (figure 2) (Experian, 2018).

`

Figure 2: Businesses confidence of their own fraud detection. Based on Experian (2018).

4.4 Banking Confidentiality

Historically, banking confidentiality was an obligation of the banks themselves. The customer information secrecy was an ethical obligation and not a legal requirement. Later, the secrecy became a matter of basic human rights. For example, Article 8 of the European Human Rights Conventions recognizes the person's private and family life as well as home and communication as private information. (Cranston, et al., 2018). As of May 2018, the EU directive 2016/679 went into effect. The directive, also known as GDPR, considered any personally identifiable data as secret. Personal identifiable data is any piece of data that can be used alone or conjointly to identify a person. This means that even fraudsters' data is protected. GDPR makes implementing a fraud database harder for a start-up.

`

4.5 Money Laundry

Money laundry is to masquerade the source of illegitimate money to appear legitimate. Money laundry might be as old as governments and states. It serves many fields of crime to extract the gain from the financial system. Most frequent crimes are conducted for financial reasons. While terrorists and drug traffickers are the top examples of money laundering, fraudsters also need to conduct money laundry. Unlike drug traffickers and terrorists, fraudsters are usually not on watch lists. (Sullivan, 2015 p. 7).

Authorities, such as Interpol, post notices regularly about wanted criminals. These databases are updated and managed separately. The databases include only public information. Authorities might also push secret notices to banks. Several European states have implemented a national bank account database. Lower financial criminals are usually excluded from these databases. As pointed out previously, the police ask not to report fraud crimes under certain money amount.

Money laundry is a three-stage operation. Money laundering starts with the placement. Placement is the point of entry for the cash into the system. A money launderer must start with one financial institution to deposit the cash or transfer the stolen funds. The second step is layering, where the launderer makes numerous transactions. These transactions span across and mesh several companies and entities. The layering is used by the fraudulent user to distance the money from themselves in case investigation happens. Layering efficacy is improved by adding layers. Each extra layer adds another distraction for a possible investigator. The transfer might span multiple countries and use more than one mean of transfer. Settling the money for spending after layering is the Integration, the third stage of money laundry. In integration, the money is assimilated into the normal economy. The money that started as illegitimate is then rendered into assets and commodity that appear legitimate. (Sullivan, 2015 p. 12).

`

The world is moving towards more digitalization. This digitalization is a blessing in reducing the cost of business and giving access to customers spanning multiple countries with ease. However, digitalization simplifies the layering process for money launderers. Adding more layers to a money laundering process makes it harder to detect. These layers are easier for addition if creating a bank account is fully online. A money launderer can create multiple accounts with several financial institutions with ease. Aside from the ease of signup online for financial services, there are other reasons why money laundry is moving online leveraging online financial service providers for their schemes. Online financial service lacks the physical presence requirement of a traditional bank, making online registration an attractive target for fraudsters. Identity verification is harder online comparing to face-to-face interactions. Online banking is one means of money launder. Digitalization has opened several other methods in which money could be layered. These methods include:

- E-Commerce: Starting an online shop is simple and accessible online. There are market solutions where a merchant can establish a shop, connecting that shop to payment services in minutes. This speed of founding is positive for honest merchants. However, the same speed and ease is applicable for fraudulent users and money launderers.
- Digital currency: Bitcoin, Ether and other digital currencies were designed around privacy. A crypto wallet (where digital currency assists are held), unlike a bank account, does not require identity verification. This makes digital currency a valuable choice for laundering money. A launderer can buy Bitcoin with a stolen Fiat and exchange that Bitcoin for Ether. The purchased Ether currency has no information on the source of fund aside from the wallet it belongs to.
- Social Media: Money launderers are using social media to recruit money mules. A money mule is a person that has no criminal priors, thus making it easier for the mule to transfer money. Since there is no prior criminal record, monitoring is usually lighter. The launderer instructs the mule to open multiple bank accounts. The mule then is instructed to transfer the money between these accounts. In effect, the mule is an instrument of layering. The money laundering mule will instantiate the payment, forcing their own information as payment initiator while hiding the fraudsters' information. Facebook alone has 2.7 billion active users, making the platform a perfect mule recruitment tool. (Sanction Scanner)

`

5 Building Proposal for Improving the Fraud Countermeasures for Banks

Implementing financial solutions is a troublesome process since there is a lot of complicated regulations governing the sector. The most recent related regulation is GDPR. The aim of this project is to find a way to work under the current regulations forming a fraud countermeasure.

5.1 Overview of Proposal Building Stage

Based on interviews with fintech industry professionals, the need for a fraud database was identified. This fraud database must be accessible to as many regulated entities as possible. Upon identifying the rampant fraud issue, multiple possible customers were communicated with. Contacting possible customers serves to validate the market preparedness for an anti-fraud solution. The idea was also pitched to investors for possible finance of the product. Legal firms and authorities were contacted as well to validate the legal possibility of such a solution.

The technical implementation was easier to propose. The author is working in the industry as a technical team lead for the customer due diligence in Holvi payment services. This position allows the author to work on multiple internal capabilities. The internal capabilities share the same legal and technical basics of the proposed solution.

`

5.2 Proposal

“Computers have far to go to match human strengths” (When will computer hardware match the human brain?, 1998). As such crowd sourcing has become a common way to generate data on the internet. Crowd sourcing is not prohibited to restaurants and hotels reviews. In this application, banks themselves (crowd) can generate enough data to defend the industry as a whole. During the research in this project, banking sector professionals agreed that compliance is a non-competing activity. Compliance as it seems by the industry is a cooperative activity.

The possible process to solve the fraud issue is simple. The first bank that detects a fraud case reports it directly to the national fraud database with a unique identifier per unique customer. This unique identifier can be used to aggregate multiple reports into one case. If the fraudster tries to register again with another bank, the previous case is found with ease. Finnish online services use Finnish Trust Network (FTN) to strongly authenticate their customers. The real identity of a website user can be easily confirmed. This means that a bank can get the unique identifier of a customer to use against the fraud database.

This project was meant to create a real production-quality product. Implementing a working product proved harder endeavor than expected. Keeping real customer data is restricted due to privacy laws. This restriction on data storage includes fraudster customers’ data. Implementing production ready database require a banking license to abide with privacy regulations. As going to the market required a banking license, multiple dummy solutions were tried as proof of concept. The technical implementations then are more of a proof that the database can be implemented in many ways.

One solution that was considered is hashing the user identifier. The aim of hashing the user identifier is to make the fraud database GDPR compliant.

`

However, this solution was also not legal since the hashed information can still be used to identify the customer.

The full proposal then is a service that exposes Restful API for adding and querying fraud cases based on the natural person's unique identifier. In Finland, the unique identifier of a natural person is the personal identity code (Henkilötunnus) (DVV). When a fraud case is discovered, the discovering bank creates a new case in the system with a simple payload shown in figure 3.

```
{
    "personal_identity_code": "131052-308T"
}
```

Figure 3. Simple payload to create a new fraud case.

This personal code will be hashed via the sha512 hashing function resulting in a code that is not reversible. This would allow the database to operate without having readable personal identity codes. Querying the database for the hashed identity code is a straightforward process after that. The matching results may vary based on the implementation details. In most of the tests written in this project a list of banks where this fraud was discovered is returned to the requester. This is not mandatory since more data can be returned such as the date of the fraud case or the text investigation of the fraud case.

`

5.3 Project Components

As with any engineering product, the project consists of multiple parts that work in synergy to function. In web technologies, these parts are usually an interface layer, business logic layer and data persistent layer. The fraud database must use a persistent data store since the information need to be permanent. This essential requirement affects how data storage is handled and limits the options. The business logic component is more flexible. A set of simple operational processes were chosen as a proof of concept implementation. The product can grow in complication in many directions. The decisions are left for the implementer's discretion. Mostly, the customers and the pricing module and many other factors can influence these processes. The last component is the interface. In this project, only an API implementation was tested. The graphical user interface or other types of interfaces could be implemented on top of the API components.

5.4 Business Logic

"Compliance is a non-competitive activity" mentioned by one of the interviewees. Compliance is used in the industry to describe all the legal activities a financial institute must oblige to. This includes activities such as AML activities, fighting fraud and other legal issues. Compliance is a shorthand for "compliance with regulations". This quote means that in the financial sector, banks cooperate to solve compliance issues. This quote and many interviews influenced the architecture of the system. The fraud database is built as a backend for banking cooperation. This is the reason that banks in the database schema are referred as a "partner".

As the project deals with investigations, the other important entity in the database is "case". A case is a collection of the banking fraud investigation findings.

`

This case can be the same legal case communicated with the authorities. Furthermore, a case can be information prepared by the investigator specifically for sharing with other partners. A “fraudulent user” as well as “reporter” tables are used to store the information on the real persons in the system.

Business entities (classes):

- Partner
- Case
- Fraudulent user
- Reporter

For the fraud database to work, it requires a set of processes to manipulate and interact with the data:

- Searching is the process where the database is queried for a unique personal identifier. The search process must be done with low latency. The searching functionality is predicted to be the most used function. For each new customer onboarding (or ongoing due diligence process) a search is done.
- Case addition is the process of adding a new entry to the database. This function is used when a partner discovers a fraudulent user in their system. A new entry is added to the fraudulent user store/table. A new case row is also added having the details of the fraud investigation. This case is what other partners will see when they have a hit on a customer.
- Case removal is when a partner concludes the fraud investigation negatively. This means that the customer is not a fraud case. This function might also be used when a fraud case expires due to age.

5.5 Database Selection

It is clear for the reader that the database component is the most important part of the project. After all, the word “database” is part of the project name. The database is an organized collection of data with APIs to interact with the stored

`

data. Implementation details vary wildly in databases from the hardware that is persisting the data into the structure and functionality of the database. Databases can be divided into multiple styles and genres. Getting into categorizing all types of persistent data storage is a futile effort for this thesis. The database selection will then be dropped to choices that fit the application requirements. The user identifier is unique, and the database should preserve uniqueness. The database should be fast to retrieve. A fraud database is used extensively in the client's onboarding processes. These processes should not be delayed by a third-party compliance service provider. These two requirements can be achieved by a range of databases. The first and most obvious genre is the relational model. Relational databases are a tried and tested choice. Relational databases originated as means to store ledger. The relational database model satisfies both conditions easily. The basic feature of the relation model is ACID compliance. ACID is short for atomicity, consistency, isolation and durability. The model also helps starting the project without the need to think about all the ways the data will be queried.

`

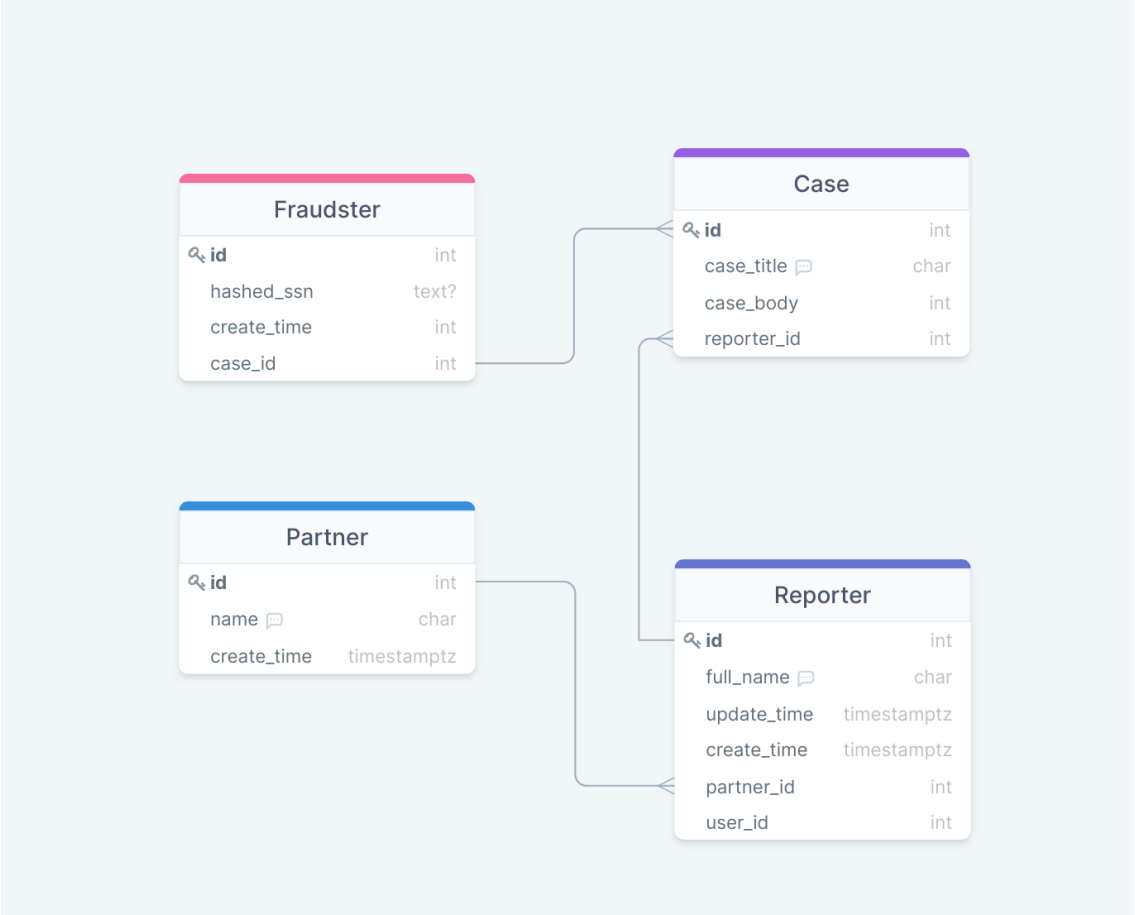

Figure 4. A drawing of the data models in a relational database

As seen above, the data model is simple. This design could be extended to include other models. For example, a user model to store the details of the user login. A set of models can be added to introduce authentication and authorisation. One possibility is using an ORM to interact with the database. ORMs are considered now to be an industry standard although they have had to overcome a historical bad reputation. The first ORMs were susceptible with SQL injection exploits (Wadas, 2016). However, it is well understood now that an ORM is better in sanitizing user input. User input sanitization is important in preventing the main attack vectors of SQL injection (Podjarny, 2016).

5.5.1 NOSQL

The Not Only SQL database came to envelop all types of databases that does not follow the Relational Database Model RDBM. This includes a variety of databases like graph databases and key-value (KV) databases. Each type has its own unique properties that can drive the database choice. Velocity DB, for example, is a database solution that advertises the data retrieval speed.

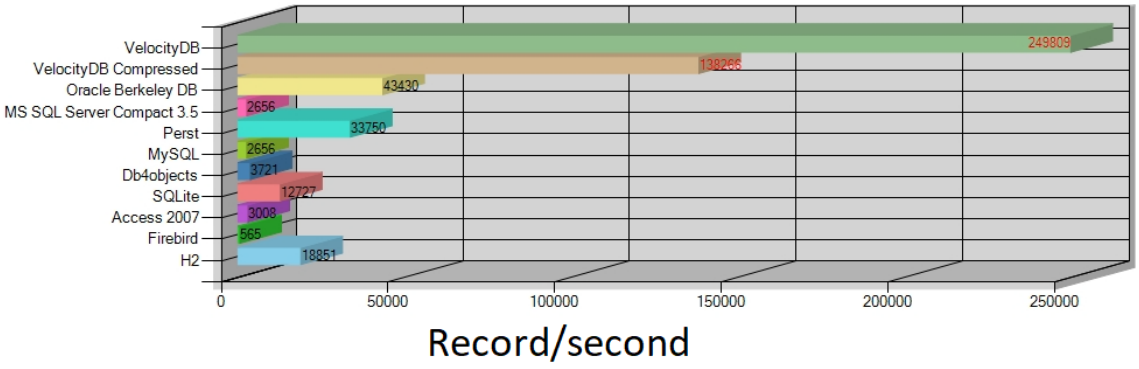

Figure 5: Record addition speed comparison between multiple DB engines (VelocityDB, 2021)

The figure above describes the speed comparison between multiple databases engines. The speed comparison in figure 5 is for item addition to DB (VelocityDB, 2021). However, there is a caveat. This figure compares the record addition speed, which is irrelevant to the fraud database. Nonetheless, this table points clearly to NoSQL solutions having an edge in certain aspects.

In the fraud database, ACID compliance is the main criteria of picking the database component. Atomic transactions are more important than scan speed. In an application with compliance, the reliability of the data is more important than the query speed. However, fast data scanning and filtration could be achieved with multi layered design.

`

5.5.2 Other Databases

Other database paradigms were considered as well. The Object database as well as the Graph database and the Ledger database. The Object database, for example, represents the data in the same way an object-oriented (OO) database does. After instantiating a class or struct, the object lives in memory with its' properties and methods. These databases use the same language to deal with data as the language the data is manipulated in the first place. Pythons' OO database will use Python code for database operations. Using python code is opposite to SQL databases where operations need to be interpreted to intermediary SQL syntax. OO databases have features that other databases cannot match such as native cache validation.

Furthermore, it provides easy testing and pluggable layered storage. The easy testing is caused by the database using the same language used to write the application itself. Pluggable storage allows the database to use any style of storage backend, such as in-memory stores easily. Despite the features that make OOPDB an attractive choice, the reality is that it is not fitting for a fraud database application. CEDD models are simple, and OOPDB is optimised for highly related objects (ZODB, 2020).

Ledger databases offer the unique ability to provide distributed trust. Ledger databases are connected to the famous crypto currency Bitcoin. While the name ledger DB might attract investors' eyes, the comparison in this thesis is focused on practical reasons to pick a database choice. Ledger DBs are oriented into distributed trust. A fraud database for financial service providers is at the core of the centralized trust domain. The national fraud database is a read operation heavy, a capability that the ledger API does not excel at. On the other hand, operating a ledger DB is less efficient in power usage due to high redundancy.

Biology, economy and several other domains use graphs to represent data. Using graph structures in computers resulted in graph databases. Graph databases excel at presenting deeply connected data. While some graph databases use relational databases as a persistent layer, other databases store the data

`

on desk in a graph format. Packing data in graph format on desk allows graph databases to store connected data closely. This in effect enhance the data retrieval speed of related data. Furthermore, a graph is the most white-board friendly database. An intuitive white board design draft is a match to the graph DB design. Graph databases are, however, unsuitable for the national fraud database. Graph databases are suitable for sparsely connected data (Ian Robinson, 2015 p. chapter 3). The national fraud database is simply connected data. For the basic data model, a limited amount of connections is set. Due to the data model, the graph database is not suitable either. In the future, a graph database can aid to represent cases. Fraud cases are usually connected in a network behavior. A successful model can take into consideration all entities related to a case in a graph shaped data. A possibility of early threat detection might arise. Early threat detection is a good feature that can be developed separately.

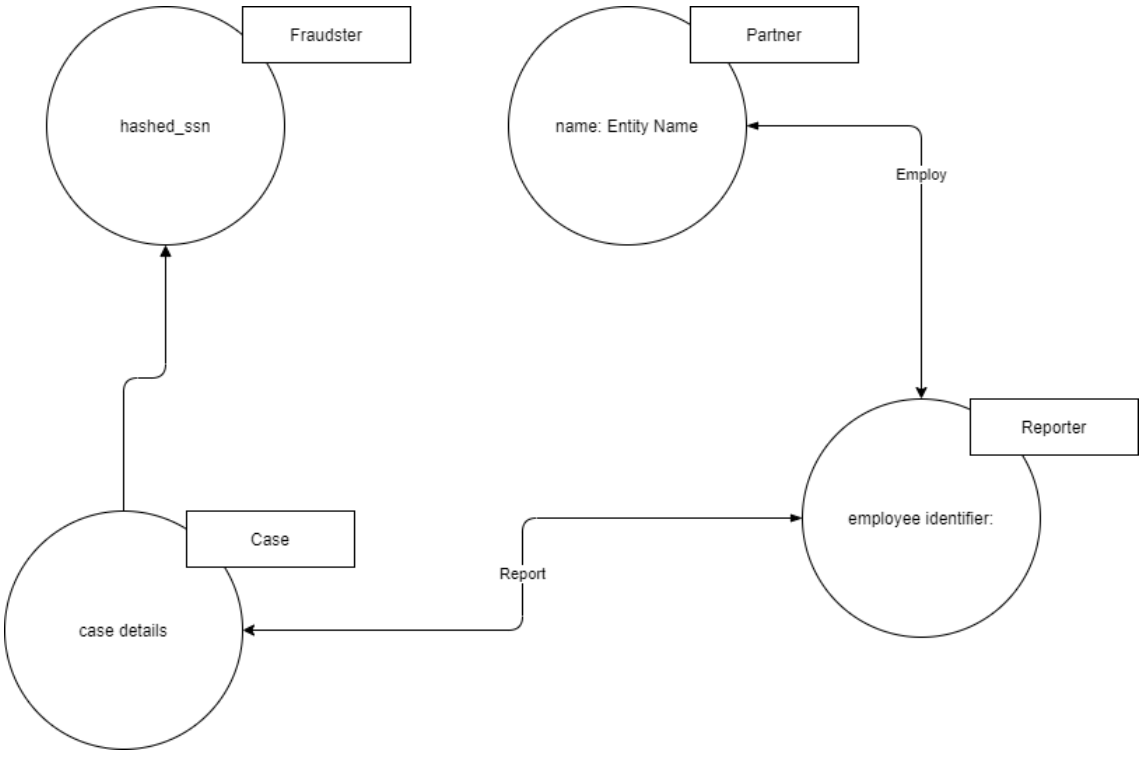

Figure 6: Graph representation of fraud database

`

In figure 6, a diagram similar to a white board sketch emerges. One of the strengths of the graph database is that the model is close to physical life representation of the data model. A graphQL compliant query for the database is presented below:

```
(f:Fraudster {"hashed_ssn": "sha512 hexadecimal code")
<-[:Investigate]-(c:Case)
<-[:reported]-(r:Reporter)-[:employee]->(p:Partner)
Return f, c, r, p
```

Figure 7. Graph QL query

As can be seen in figure 7, graph QL code is intuitive and bares clear resemblance to the representation in figure 6.

`

5.5.3 Suitable Data Design

Even with rising fraud attacks, most of the banking customers are legitimate non-fraudulent customers. This will reflect on the number of hits. Most scan/search queries will find no match. This means the number of records is going to be low compared to number of requests. Analyzing the data, an axis could be identified to split the data. The hashed SSN (the search operation key) is a sort key that can be used as a database table index. The database for the hashed SSN can be an in-memory Key-Value database like Redis. Redis supports fast retrieval of data. A hashed SSN can be stored in Redis as the lookup key, the value could be the unique identifier of the case in the relational database. This means that a fast Redis query would result in finding the connected case, which has all relevant details. The SSN identifier is hashed with sha256 which results in 64 hexadecimal characters. This small data size fits in a server with limited memory size (RAM).

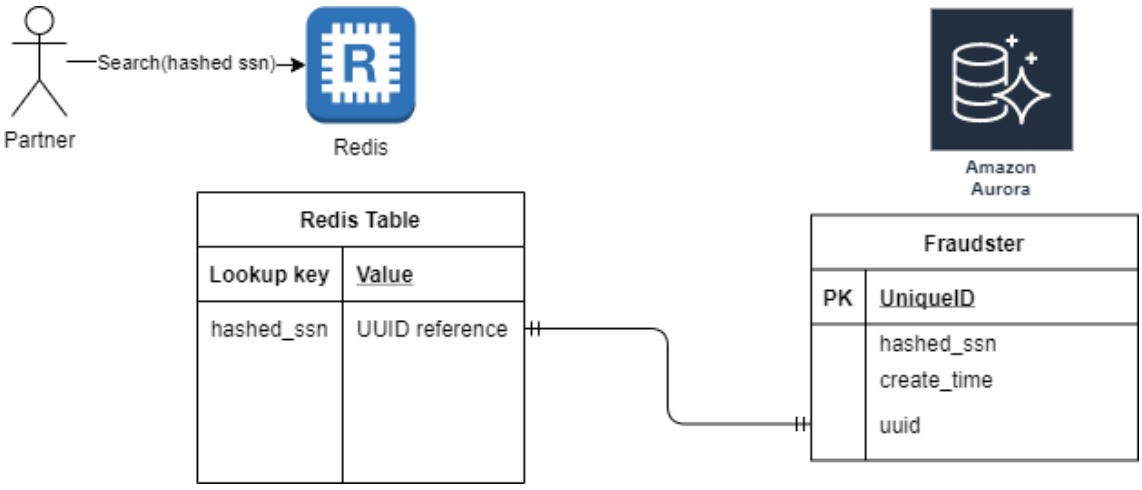

Figure 8. Design for a two-layer approach

`

Figure 8 shows a simple two-layer approach in the database. This approach leverages the strong point of each database paradigm. The SQL database layer will ensure the consistency of sensitive data through the ACID compliant persistence. The Redis layer will reduce service latency and increase reliability under large loads.

5.6 Interface

For most users, interfaces themselves are the product. A product interface is the easiest product component to comprehend by a user of an application. The best interfaces are designed with their user in mind. Interfaces have varied since the invention of computers, from the first versions of punch cards into the latest attempts of neural links. While a direct link to the human brain is appealing, it is probably not the best interface yet for a fraud database application. Considering the usage of the product, the most used functionality is the search functionality. The search functionality needs to be called upon each customer onboarding. The main trigger of the search functionality is the partner's onboarding/signup server. A manual search could also be conducted although manual searching is not feasible for organizations with thousands of daily onboarding applications. Therefore, most searches are triggered by the partner's server directly and automatically to the fraud database. The last fact leads by nature for an API (Application programmable interface). A graphical user interface (GUI) via the web is recommended but not mandatory for the first version.

For an API, many solutions could be used. Most of the banks use an old "SOAP" interface supporting legacy systems. Suomen Asiakastieto (SAT) is one of the most important vendors of a customer's financial data in Finland. SAT migrated from the legacy soap interface only recently (in 2017). What is called "web 2.0" uses instead an implementation named "restful API" for programmatical communication. Other API designs were considered such as Graph API. Graph APIs are designed based on Graph theory with the aim of representing

`

deeply connected data. Restful APIs is the technology focused on in this research (and prototype).

Along with the previous requirements, the following requirements influenced the decision of using restful API for the fraud database interface:

- Modern technology: The popularity of the RPC interfaces is dropping. Most banks keep supporting soap interfaces due to their legacy banking core systems. The popularity of the design is important for long term maintainability. The modernity of technology is also important for the software dependencies. One of the main libraries for SOAP in Python dropped support recently due to lack of maintainers.
- Common usage: The main users for the API are developers writing consumer applications. A ubiquitous API design means an easier relationship with the client development team with cheaper integration cost.
- Concise design: Some API designs provide unique features that can be useful with complex applications. Graph QL for example provides a flexible interface to query complex data relationships. Graph APIs are then beneficial with deeply connected entities. In this project's application, the data structures are simple with discrete direct connections. For the first version of the fraud database, Graph APIs are a complication that is not worth the cost.

As a web product, the choices for web APIs are mainly stable and agreed paradigms. Paradigms such as Representative State Transfer REST which leverages the web's verbs to interact with the data are well studied. Other paradigm choices such as Remote Procedure Call RPC which communicate via remote code execution are also considered. The final option considered is GraphQL.

`

5.6.1 REST API

REST APIs are designed with the resource in mind. Hence, resources are part of the URL structure. GET, POST, PUT and DELETE are the HTTP verbs used by the REST paradigm to perform actions on the resource. The POST method is used for creating a new instance of the resource. The GET method is for reading an object. PUT is for updating a previous resource. Delete is for deleting a resource. To communicate the interface details, an open API documentation specification is used. One of the top used specification is Swagger. There is large support for swagger and as such, it was used for the REST documentation. Swagger documentation can be used to generate the client's applications in most programming languages. An Open API specification is attached to this project.

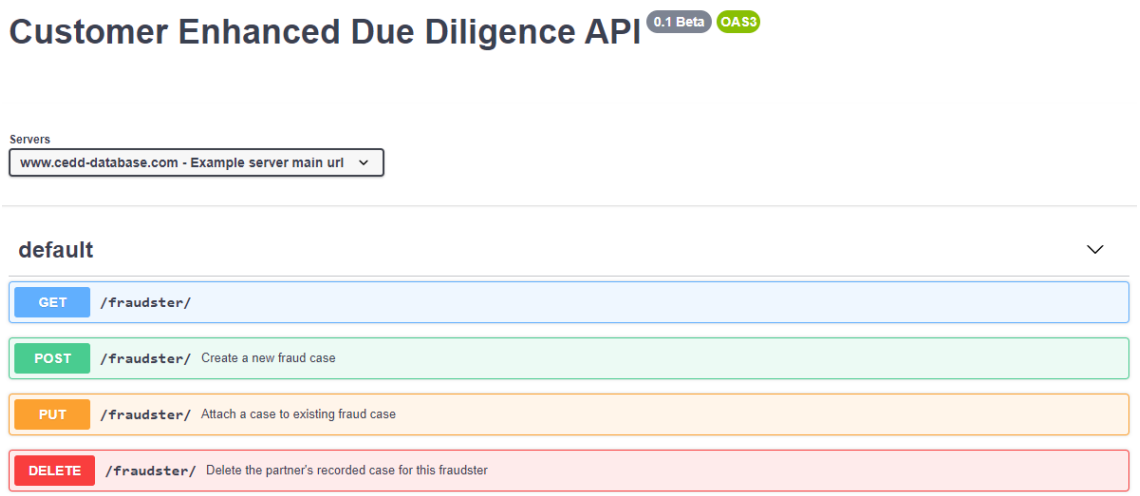

Figure 9. A render of the swagger API specification

The figure shows a render of the CEDD REST interface. A web tool called https://editor.swagger.io/ was used to render the specification. In the Restful API design, four methods were used to interact with the endpoint. A partner can

`

search, add, update and delete cases using the API. Swagger could also be used to generate client and server code alike.

5.6.2 Remote Procedure Call

Remote procedure call (RPC) was introduced as a web service approach. The aim of implementing RPC is to simplify calling a procedure from remote computers. RPC has an emphasis on reusing procedures that originally developed for humans. Furthermore, RPC employs the use of the XML file format to describe the procedures. The XML format is used for transport layer, where the transported data is passed to the underlying handler function. Just like calling a function in local code by direct invocation, RPC allows remote invocation of functions. Most programming languages have an RPC framework that can help expose functions to remote invocation with ease.

RPC predates the web itself, as such, the paradigm has morphed to be one of the most important web API development approaches. (Wilder-James, et al., 2001).

The REST API design focuses on the “resource” being the main subject of the API. Alternatively, RPC focuses on the process. RPC has two main components, the procedure (see function, method) and the parameters. The parameters are the data points used for the function invocation. To simplify intra-systems communication, RPC data is typed. RPC itself defines a set of simple data types with the ability to extend to complex developer-defined datatypes.

Recalling earlier parts, the main processes of the fraud database are:

- Search: To search previous cases associated with a fraudulent user.
- Add: To report a new case associated with the user as well as the reporting partner.
- Delete: To remove a previously added report.

`

Each of these three processes has an entry point in the program. The entry point is normal function.

```
def search_by_hashed_ssn(hashed_ssn: HashedSSN):
    """Search for a fraud by social security number

    Args:
        hashed_ssn (HashedSSN): The hashed SSN of the fraudlent user.
    """
    # logic here
    pass
```

Figure 10: Python code for the search function entry point.

In figure 10, the RPC implementation is exposing the search function for remote callers. The RPC frameworks can also expose the HashedSSN data type for the remote callers for client-side validation. Frameworks like Python's Spyne can be used to expose these process functions (Spyne). The framework can also generate XML schemas for these procedures. Consequently, the generated XML schemas can be used to generate client consumer code with ease.

5.7 Monetization

This project is describing a commercial product. Commercial products need to earn money. Products such as online fraud databases are called Software as a Service (Saas). This means a customer needs not to worry about the infrastructure and maintenance cost. Instead, the customer pays for the usage right of the service.

Pricing online products can be categorized into two groups: fixed pricing and dynamic pricing. Fixed pricing, also called static pricing is the model where the product cost is not affected by state or circumstances. This pricing has the benefit of assuring customers of the service since service details are predefined.

`

Meanwhile, dynamic pricing (also called real time pricing) has flexible cost varying with time and conditions. Fixed pricing is logistically easier to implement. Dynamic pricing requires calculation of resources used and the time scale of usage. Hence, dynamic pricing diverts scarce human resource from developing the product features into managing pricing and fees. (Pricing schemes in cloud computing: a review, 2017 p. 3).

- Looking at companies working in the same domain (user information database):
- Asiakastieto: The Finnish consumer information database is priced per request. The price of each request depends on the volume of requests, with the price decreasing in opposition to request volume increase.
- ComplyAdvantage: Comply advantage has a fixed price per partner. The fixed sum per customer payment scheme is known in the industry as “eat all you can”.

Contemplating the pricing models, for CEDD, the best approach is the dynamic pricing model. A fixed pricing scheme would not be fair for the partners or the provider. Partners with low request volumes should not pay as much as partners with high volumes. While Nordea (the biggest bank in the Nordic countries) will generate thousands of requests per day, other smaller partners might have a low daily request count of tens. As described earlier, the dynamic model requires inflated cost of development and maintenance. The dynamic cost calculation can be simplified by charging per number of requests. Partners will pay as per their usage. The/A discount could be made for large volumes.

Total monthly price = cost per request * number of requests

`

As the source of data is the partners themselves, the database value can be increased by increasing the amount and quality of the data. Increasing the quantity of data in the database could be achieved by implementing price discount strategy. Adding fraud cases requires extra time/energy investment from the partner to prepare the data for CEDD storage. This cost of addition could be balanced with positive credit per case addition. The positive credit worth is not going to complicate the pricing calculation drastically.

Total monthly price = (cost per request * number of requests) – (number of additions * negative credit returned)

5.8 Ongoing Due Diligence

An important feature that can be added to CEDD is ongoing due diligence. Each time a partner is searching for a hashed SSN, the hashed SSN could be stored in a separate database in relation to the searching partner. If the hashed SSN is not found in the database, it is still stored in the ongoing due diligence database. When a hashed SSN is added to the fraud database, a notification webhook can be sent to the partner informing with the new case. This is to fight fraud attempts to create accounts, then using them for money laundering at a later date. Upon the date of account creation, there is no fraud and the customer is legitimate. When the first fraud is detected by another partner, all partners that have active accounts for this customer are alerted. This feature can be a strong protection method. Meanwhile, the fraud database can sell this ongoing due diligence solution as an extra product.

`

6 Validation of the Proposal

As the product relies on crowd sourcing of fraudsters' data, the data can only be accessed by regulated services; furthermore, the data can be added only by regulated services. Hence, the fraud database has a short list of possible customers limited with licensed financial services. Nonetheless, the fraud database is needed in the market with no similar product filling the space. Fraud is running rampant in the European Union with little tooling to avoid fraud risk. The losses are visible, and the lack of tooling is identified by the financial industry as well as the authority.

The idea of shared trust between banks is not novel. Finnish banks have a long history in cooperation. This project simplified the requirements by storing identifiers only without much details about the investigation and cases. This simplified setup is not sufficient for real life applications, with case category being an essential piece of information. Thus, the database requires at least hit categorization.

This database is a product that is too small to sustain a company. The fraud database requires a financial service license to operate. Licensing fees as well as legal back-office requirements would make a standalone fraud database not sustainable. As such, this fraud database is suitable to be a complementary offering to other AML tooling. Several companies operate in AML and onboarding, which means that there is a whole new field called "Regtech". Companies like SAT are a good candidate to operate a fraud database. SAT has portfolio of registry information solutions where a fraud database fits nicely.

Data quality is identified as the most important aspect of a fraud database. Competitors like CIFAS have mandatory training for all new partners of their fraud database. The training includes a representative from CIFAS going through printed materials with employees of the new onboarded partner. During the onboarding period of a new partner, this partner's entries to the system are ranked as lower quality. After a review period that includes manual reviews of

`

the input, the new partner becomes a full member. This level of diligence is necessary to preserve a high data quality.

The profitability of the fraud database is not instantaneous. Deductions of cost will be required to incentivize partners to use the system. After the initial period of discount to collect fraud data, the pricing can be started. Price tiers are essential for API economy, and customers must have discount on large usage. This discount encourages companies to grow their usage of the product, allowing more data to flow. Competitors like CIFAS request the users to fill a monthly report with fraud caught using the platform. The fraud must include estimation of the money saved by using the fraud database. CIFAS then shares those saved funds with the customer by charging a percentage.

AML is a domain with fast paced regulatory changes. Due to rapid regulation change, the companies working in the field are required to be nimble. Hence, users in financial technology domain require their vendors to be on the same level of technological agility. Other technological requirements include having a good documentation for the API. In this project, the most ubiquitous type of API documentation was used.

`

7 Summary and Conclusions

The aim of the project was to implement a fraud database as a real-life product. The product is to provide information for the financial domain. Solutions that introduce technology into the financial domain are called Fintech. The first part of this chapter is a summary of the work and findings of this project. The resulted proposal consists of designs and specification. The implementation of the plans is discussed in the second part of this chapter. Aspects spanning from legal to technical were analysed in this project. The last part of this chapter is a retrospect of the author's journey.

7.1 Executive Summary

Being a FinTech solution, the fraud database functions in a regulated domain. The first assumption of this work was the possibility of overcoming the regulatory restraints with engineering. The hypothesis is that a database operator that is ignorant of the user identifier can operate without licencing. GDPR aims to protect natural persons within the EU. If the fraud database could not identify natural persons, a regulation concerned with people would not apply. This reflected on the design of the database having the hashed persona identifiers as the key for search in the system.

During the research, a legal firm was approached to clarify the legality of operating under the suggested design. The law firm explained that even with the database operator unable to identify the persons in the system - due to irreversible hashing - a third party can. This meant the product is impossible under GDPR with its only exception being regulated entities. Having a banking license can allow this database to work legally as part of the financial system.

Having a banking license requires energy that a modern start up cannot afford. The licensed entities have duties and obligations towards the government. Aside from an expensive stand-alone product, other setups exist. The fraud database could be operated by a pre-existing licensed banking services. A more

`

suitable match for a fraud database product can be found in registry-techs. Registry techs are licensed entities and they have similar product offering where a fraud database can fit.

7.2 Next Steps and Tips for Implementation of the Proposal

The heaviest load in implementing this product is the legal requirements. Overcoming the legal requirements requires huge resources. This product has a niche market with profit that might not cover the legal requirement. The technical part is simple with basic operations outlined in this research. The design was intentionally non-intrusive to ease customisability.

7.3 Final Words

Engineering is a creative endeavour. This research started with a hypothesis that unfolded into discovery of knowledge. The author acquired knowledge such as fraud intricacies and fraud countermeasures. Technical aspects were studied to create a database by comparing different technologies. This comparison of technology resulted in forming deeper understanding of these technologies.

`

References

Central Bank Of Malysia. 2010. What Is Financial Fraud. [Online] 2010. http://www.bnm.gov.my/microsites/fraudalert/01_what.htm. [Cited: 10 February 2019.]

CIFAS. About-Us. [Online] https://www.cifas.org.uk/about-cifas/what-is-cifas. [Cited: 12 04 2021.]

Cifas. 2018. *Fraudscape 2018.* London: Cifas, 2018.

Sir Ross Cranston, Emilios Avgouleas, Kristin van Zwieten, Christopher Hare, and Theodor van Sante. 2018. *Principles of Banking Law.* Third. OUP Higher Education Division, 2018.

DVV. Personal Identity Code. [Online] https://dvv.fi/en/personal-identity-code. [Cited: 21 October 2020.]

EBA European Banking Authority. 2018. *Final Report on EBA GL on Fraud Reporting.* 2018. EBA/GL/2018/05.

European Central Bank. 2018. *Fifth Report on Card Fraud.* Frankfurt am Main: European Central Bank, 2018. ISBN 978-92-899-3192-2.

Experian. 2018. *The 2018 Global Fraud and Identity Report.* Experian, 2018.

Federal Deposit Insurance Corporation. Your Rights to Financial Privacy [Online] https://www.fdic.gov/consumers/privacy/yourrights/index.html. [Cited: 12 April 2021.]

Finnish Competetion and Consumer authority. 2019. Competition Control and Advocacy. [Online] July 2019. https://www.kkv.fi/en/current-issues/press-releases/2016/28.11.2016-fccas-decision-narrows-banks-possibilities-to-increase-charges-and-fees-for-consumer-credit/. [Cited: 21 April 2021]

Finnish Parlement. 2013. Act on Detecting and Preventing Money Laundering and Terrorist Financing (503/2008. *Finlex.* [Online] 2013. https://www.finlex.fi/fi/laki/kaannokset/2008/en20080503_20130327.pdf. [Cited: 12 April 2021]

FINTRAC: Financial Transactions and Reports Analysis Centre of Canada. 2017. Large Cash Transactions. [Online] 6 29, 2017. http://www.fintrac-canafe.gc.ca/reporting-declaration/Info/rptLCTR-eng.asp. [Cited: 2 October 2019.]

Gee, Sunder. 2014. *Fraud and Fraud Detection.* s.l.: John Wiley & sons, Incorporated, 2014.

Ian Robinson, Jim Webber, Emil Eifrem. 2015. Graph Databases. s.l.: O'Reilly Media, Inc., 2015.

`

Jan Putnis, Tamara Raoufi and Jennyfer Moreau. 2021. The Banking Regulation Review: European Union Overview. *The law review.* [Online] https://thelawreviews.co.uk/title/the-banking-regulation-review/european-union. [Cited: 12 April 2021.]

KPMG international. 2019. Global Banking Fraud Survey. [Online] 2019. https://assets.kpmg/content/dam/kpmg/xx/pdf/2019/05/global-banking-fraud-survey.pdf. [Cited: 12 April 2021.]

Podjarny, Guy. 2016. snykblog. [Online] 2016. https://snyk.io/blog/sql-injection-orm-vulnerabilities/. [Cited: 14 January 2021.]

Soni, Aishwarya and Hasan, Muzammil. 2017. *Pricing schemes in cloud computing: a review.***.** 2277-7970, 2017, International Journal of Advanced Computer Research,, Vol. 7(29).

Sahlin, Philip. 2009. *Luxembourg Banking Secrecy; Privacy tool or fraud facilitation? .* s.l. : FACULTY OF LAW; University of Lund, 2009.

Samociuk, Martin and Lyer, Nigel . 2010. *A short giude to Fraud risk : Fraud Resistance and Detection.* s.l. : Routledge, 2010.

Sanction Scanner. The Change of Money Laundering in The Digital Age. [Online] https://sanctionscanner.com/blog/the-change-of-money-laundering-in-the-digital-age-190. [Cited: 23 January 2021.]

Spyne. Spyne. [Online] http://spyne.io/. [Cited: 23 January 2020.]

statista. 2021. Digitial Payments. [Online] https://www.statista.com/outlook/296/100/digital-payments/worldwide?currency=usd. . [Cited: 31 January 2021.]

Sullivan, Kevin. 2015. *Anti-Money Laundering in a Nutshell: Awareness and Compliance for Financial Personnel and Business.* s.l.: Apress, 2015.

The Law Dictionary. The Law Dictionary. [Online] https://thelawdictionary.org/fraud/. [Cited: 10 February 2019.]

The United States Congress. 1970. Bank Secrecy Act. [Online] 1970. https://www.ffiec.gov/bsa_aml_infobase/documents/fdic_docs/bsa_manual.pdf. [Cited: 10 February 2019.]

Tilastokeskus. 2020. Justice statistics. [Online] http://www.stat.fi/tup/suoluk/suoluk_oikeusolot_en.html. [Cited: 14 September 2020.]

VelocityDB. 2021. VelocityDB Compare. [Online] 2021. https://velocitydb.com/Compare. [Cited: 14 January 2021.]

`

Wadas, Michał. 2016. SQL for programmers-indexes are not magic. [Online] https://applandeo.com/blog/sql-programmers-indexes-not-magic/. [Cited: 14 January 2021.]

Moravec, Hans. 1998. Journal of Evolution and Technology 1998. *When will computer hardware match the human brain?*

Wilder-James, Edd, et al. 2001. *Programming Web Services with XML-RPC.* s.l.: O'Reilly Media, Inc, 2001.

ZODB. 2020. ZODB documentation. [Online] http://www.zodb.org/en/latest/. [Cited: 16 January 2021.]

`

Appendix 1

Django code for fraud database main models:

```
from django.contrib.contenttypes.fields import GenericForeignKey
from django.contrib.contenttypes.models import ContentType
from django.db import models
from django_fsm import FSMField
from django_fsm import transition
from core.constants import STATE_ACTIVE
from core.constants import STATE_INACTIVE
from core.constants import STATE_PENDING
from core.constants import STATE_TERMINATED
from core.constants import LOG_ACTIVITY_CHOICES
from core.constants import ACTIVITY_CREATE
from core.constants import ACTIVITY_CREATE
from core.constants import ACTIVITY_DEACTIVATE
from core.constants import OPERATIONAL_STATE_CHOICES


class BaseClass(models.Model):
    create_time = models.DateTimeField(auto_now=True)
    update_time = models.DateTimeField(auto_now_add=True)
    uuid = models.UUIDField()
    class Meta:
        abstract = True


class DefaultStateFieldModel(BaseClass):
    state = FSMField(
        default=STATE_ACTIVE,
        verbose_name='state',
        choices=OPERATIONAL_STATE_CHOICES,
        protected=True,
    )
    class Meta:
        abstract = True

    @transition(field=state, source=[STATE_PENDING, STATE_INACTIVE], tar-
get=STATE_ACTIVE)
    def activate(self):
        Log.objects.create(content_object=self, message='Object created', ac-
tion=ACTIVITY_CREATE)

    @transition(field=state, source=[STATE_PENDING, STATE_ACTIVE], tar-
get=STATE_INACTIVE)
    def deactivate(self):
        Log.objects.create(content_object=self, message='Object Deactivated',
action=ACTIVITY_DEACTIVATE)


class Log(BaseClass):
    message = models.TextField()
    content_type = models.ForeignKey(ContentType, on_delete=models.CASCADE)
    object_id = models.PositiveIntegerField()
    content_object = GenericForeignKey('content_type', 'object_id')
    action = models.CharField(max_length=64, choices=LOG_ACTIVITY_CHOICES)
```

`

```
from django.contrib.contenttypes.fields import GenericForeignKey
from django.contrib.contenttypes.models import ContentType
from django.db import models
from django_fsm import FSMField
from django_fsm import transition
from core.models.operational_models import DefaultStateFieldModel
from core.models.operational_models import BaseClass
from core.constants import STATE_ACTIVE
from core.constants import STATE_INACTIVE
from core.constants import STATE_PENDING
from core.constants import STATE_TERMINATED
from core.constants import LOG_ACTIVITY_CHOICES
from core.constants import OPERATIONAL_STATE_CHOICES


class Institution(DefaultStateFieldModel):
    name = models.CharField(max_length=128)


class Customer(DefaultStateFieldModel):
    code = models.CharField(max_length=512)
    institution = models.ForeignKey(to=Institution, on_delete=models.PROTECT)


class CustomerInstitutionRelationship(BaseClass):
    customer = models.ForeignKey(to=Customer, on_delete=models.PROTECT)
    state = FSMField(
        default=STATE_PENDING,
        verbose_name='Relationship state',
        choices=OPERATIONAL_STATE_CHOICES,
        protected=True,
    )

)
```

`

```
STATE_ACTIVE = 'active'
STATE_INACTIVE = 'inactive'
STATE_PENDING = 'pending'
STATE_TERMINATED = 'terminated'
ACTIVITY_CREATE = 'create'
ACTIVITY_EDIT = 'edit'
ACTIVITY_DEACTIVATE = 'delete'

LOG_ACTIVITY_CHOICES = (
    (ACTIVITY_CREATE, 'Create'),
    (ACTIVITY_EDIT, 'Edit'),
    (ACTIVITY_DEACTIVATE, 'Deactivate')
)

OPERATIONAL_STATE_CHOICES = (
    (STATE_ACTIVE, 'Active'),
    (STATE_INACTIVE, 'Inactive'),
)

CONTRACT_STATE_CHOICES = (
    (STATE_ACTIVE, 'Active'),
    (STATE_INACTIVE, 'Inactive'),
    (STATE_PENDING, 'Pending'),
    (STATE_TERMINATED, 'Terminated')
```

`