

卒業論文 2025 年度 (令和 7 年)

コネクティッドカーにおける運転歴データの管理・提示システムの構築

慶應義塾大学 環境情報学部
仁戸田晃

卒業論文要旨 - 2025 年度 (令和 7 年度)

コネクティッドカーにおける運転歴データの管理・提示システムの構築

要旨

キーワード:

1. コネクティッドカー, 2. デジタルアイデンティティ

慶應義塾大学大学 環境情報学部
仁戸田晃

Abstract of Bachelor's Thesis - Academic Year 2025

Building a System for Managing and Presenting Driving History Data in Connected Cars

I WILL WRITE ENGLISH.

Keywords :

1. Connected Car, 2. Digital identity

Keio University Faculty of Environment and Information Studies
Akira Nieda

目次

第1章 序論	1
1.1 はじめに	1
1.2 本論文の構成	2
第2章 背景	3
2.1 自動車の情報化	3
2.2 コネクティッドカーとテレマティクス	3
2.2.1 コネクティッドカーとテレマティクスの概要	3
2.2.2 テレマティクスの活用事例	4
2.2.3 コネクティッドカーの発展と展望	4
2.3 車両データを用いるサービス	4
2.3.1 利用ベース保険 (UBI)	4
2.3.2 カーシェアリングサービス	5
2.4 本章のまとめ	5
第3章 基盤技術	6
3.1 デジタル証明書と構成技術	6
3.1.1 デジタル証明書	6
3.1.2 暗号学的ハッシュ関数	7
3.1.3 公開鍵暗号	7
3.1.4 デジタル署名	7
3.2 Verifiable Credentials と Decentralized Identifiers	8
3.2.1 Verifiable Credentials	8
3.2.2 Decentralized Identifiers	9
3.2.3 選択的開示	11
3.2.4 ウォレットアプリケーション	11
3.3 本章のまとめ	11
第4章 本研究における問題定義	12
4.1 既存システムにおけるデータの流れ	12
4.2 既存システムの問題点	12
4.2.1 データのサイロ化	13
4.2.2 データのコントロール権の欠如	13

4.2.3	データの真正性の担保	13
4.2.4	各事業者のポリシーの相違	14
4.3	本章のまとめ	14
第5章	提案手法	15
5.1	提案手法の概要	15
5.2	全体構造	15
5.2.1	システムアーキテクチャ	15
5.2.2	データおよび処理の流れ	17
5.2.3	処理アプリケーション	18
5.2.4	連携アプリケーション	19
5.2.5	ウォレットアプリケーション	19
5.2.6	検証者アプリケーション	19
5.3	データモデル	19
5.3.1	基本設計	19
5.3.2	データの構造	20
5.4	システムの満たすべき要件	20
5.4.1	データのコントロール可能性	21
5.4.2	データの真正性の担保	21
5.5	前提条件	21
5.6	本章のまとめ	22
第6章	実装	23
6.1	実装の概要	23
6.1.1	技術スタック	23
6.1.2	データモデル	23
6.2	各アプリケーションの実装	24
6.2.1	自動車アプリケーション	24
6.2.2	モバイルアプリケーション	24
6.2.3	検証者アプリケーション	25
6.3	本章のまとめ	25
第7章	評価	26
7.1	評価内容	26
7.2	システムの定性評価	26
7.2.1	データのコントロール可能性	26
7.2.2	データの真正性の担保	26
7.3	STRIDE による脅威分析	27
7.3.1	STRIDE について	27
7.3.2	システムの DFD	27
7.3.3	特定された脅威	28

7.3.4	自動車における脅威	28
7.3.5	モバイルにおける脅威	29
7.3.6	検証者における脅威	29
7.4	本章のまとめ	29
第 8 章	結論	30
8.1	本研究のまとめ	30
8.2	本研究の課題と限界	30
	謝辞	31
	付 録 A 付録	32
A.1	実装のコードスニペット	32
A.2	STRIDE による脅威分析の結果	33

図 目 次

3.1	VCs における IHV モデルと仕組み	9
3.2	DIDs の例	10
3.3	DIDs のエコシステム	10
4.1	既存システムにおけるデータの流れ	13
5.1	提案するシステムの論理アーキテクチャ	16
5.2	提案するシステムのシーケンス図	18
7.1	提案するシステムのデータフロー図	28

表 目 次

5.1 システムの各コンポーネントが持つアプリケーションとその機能	17
5.2 システムで使用するデータとセンサ	20
6.1 使用した技術スタック	23
7.1 STRIDE における脅威のカテゴリ	27
A.1 自動車における脅威	34
A.2 モバイルにおける脅威	35
A.3 検証者における脅威	35

第1章 序論

本章では本研究の背景、課題及び手法を提示し、本研究の概要を示す。

1.1 はじめに

インターネット技術の発展に伴い、あらゆるモノがネットワークに接続されることはもはや自然なことになった。Internet of Things と呼ばれるこの概念は急速に普及しており、ネットワークと接続するモノの範囲は拡大する一方である。自動車もその例外ではなく、車両に通信機能を搭載することで、外部ネットワークと常時接続される「コネクティッドカー」へと進化してきた[?]。とりわけ、テレマティクスと呼ばれる、自動車をはじめとする移動体に通信システムを搭載し、リアルタイムでデータを外部とやり取りすることを可能にする技術の発展は、自動車の運転や利用状況に関する詳細なデータ収集を可能にしてきた。これらのデータは長らく自動車の安全運行のためのシステム、例えば緊急通報システムやカーナビゲーションシステムといった形で交通安全に寄与してきたが、現代ではこれらのデータが非常に個人的なものであることに着目し、自動車の利用者一人ひとりに最適化したサービスを提供しようとする試みも行われている。利用ベース保険やカーシェアリングサービスはその代表であり、車両から抽出したデータに基づく多様な価値提供が進んでいる。

しかしながら、現在普及しつつあるコネクティッドカーを基盤としたサービスにおいては、自動車から抽出されたデータは自動車メーカーや特定のサービス事業者のクラウドやデータセンター内に閉じた形で保管されていることがほとんどである。そのため、利用者が自身の運転データを他の事業者へ持ち出したり、サービスを乗り換える際に運転履歴を引き継いだりすることが非常に困難である、すなわちベンダーロックインに陥るという問題が発生している。また、データが分断されてサイロ構造に閉じ込められていると、利用者自身が自らの自動車から抽出されたデータへアクセスできないという問題も発生する。一方で、事業者がこのような構造を取らざるを得ない理由の一つとして、自動車から直接データを取得しない限り、データの真正性を担保できないという技術的な問題も存在する。

そこで本研究では、このような問題に対し、利用者を自動車と事業者の中間に論理的に配置してデータへのアクセスおよびコントロール可能性を持たせ、その上でデータの真正性を担保する新たな運転履歴データの管理・提示モデルを提案する。さらにこのモデルを評価するために、Decentralized Identifier および Verifiable Credentials という技術を用いたシステムを設計した上で実装し、実際に動作確認を行う。以上により、利用者自身がデー

タに対するアクセスおよびコントロールが可能でありながら、データの真正性を担保し、複数の主体間で安全かつ拡張可能な形で運転歴データの利用が可能となる仕組みの構築を目的とする。

1.2 本論文の構成

本論文における以降の構成は次の通りである。

2 章では、背景を述べる。3 章では、本研究において基盤となる技術を述べる。4 章では、本研究における問題の定義と、解決するための要件の整理を行う。5 章では、本研究の提案手法を述べる。6 章では、5 章で述べたシステムの実装について述べる。7 章では、4 章で求められた課題に対しての評価を行い、考察する。8 章では、本研究のまとめと課題、そして今後の展望についてまとめる。

第2章 背景

本章では本研究の背景について述べる。

2.1 自動車の情報化

インターネットの発展に伴い、コンピュータだけではなく、あらゆるものがネットワークで接続しあうことがもはや当たり前となった。このような、モノとモノがネットワークを介して繋がる技術は“Internet of Things”の略称であるIoTと呼ばれ、社会のデジタル化に欠かせない技術となっている。自動車も例外ではなく、ネットワークと自動車を接続することで自動車自体を情報化し、道路交通情報をシステム化することができる。道路交通に関する総合的な情報通信システムは高度道路交通システムと称され、“Intelligent Transport Systems”の略称であるITSと呼ばれる。自動車の情報化は社会全体の利益に繋がり、その必要性は高い。例えば、複数の自動車からデータを集めて分析することで、渋滞の緩和に役立てようという試みはその一例である。自動車の自動化は、自動車と情報通信に関連する分野に貢献することで、情報通信社会を支援する役割も期待されている。

2.2 コネクティッドカーとテレマティクス

本節では、コネクティッドカーとそれを支える技術であるテレマティクスについて概説し、その活用事例及び発展の展望を紹介する。

2.2.1 コネクティッドカーとテレマティクスの概要

コネクティッドカーとは、外部ネットワークと接続している自動車であると定義される。自動車に通信システムを搭載することで、リアルタイムに外部とデータのやり取りを行うことができ、このように移動体に通信手段を搭載し、外部と情報をやり取りする技術をテレマティクスと呼ぶ。なお、テレマティクスはIoTの一種であり、「テレマティクス (telematics)」という語は「テレコミュニケーション (telecommunication)」と「インフォマティクス (informatics)」を組み合わせた造語である。コネクティッドカーは、テレマティクスによって外部ネットワークに接続している自動車そのものである。テレマティクスに対応したコネクティッドカーにはTCU (Telematics Control Unit = テレマティクス制御ユニット) と呼ばれる部品が搭載され、TCUは自動車のECU (Electronic Control

Unit = 電子制御ユニット) と接続し、外部ネットワークとの通信を行う。また、コネクティッドカーでは“Over The Air (OTA)”という技術を用いることで車載ソフトウェアの更新を無線ネットワーク経由で行うことができる。

2.2.2 テレマティクスの活用事例

テレマティクスを用いることで、様々なシステムが実現している。例えば、自動車に関係する事故や事件などの非常事態が発生した際に、警察や救急、メーカーや保険会社などに緊急で通報する緊急通報システムや、車両が盗難された時に位置情報を元に追跡が可能な車両盗難追跡システムなどである。また、テレマティクスを用いるサービスも多く存在し、代表的なものとして利用ベース保険が挙げられる。利用ベース保険の詳細については後述する。

2.2.3 コネクティッドカーの発展と展望

コネクティッドカーは車両の通信機能によりリアルタイムで外部ネットワークとのデータのやり取りを行い、安全性と利便性、そしてシステム・サービス間の連携を大きく向上させてきた。テレマティクスを用いるシステムやサービスの利用拡大に伴い、自動車が「移動するデジタル端末」として機能しており、今後は自動運転や EV、そして MaaS (Mobility as a Service) との統合が進み、車両データを基盤とした新たなエコシステムの中心的役割を担うことも期待される。コネクティッドカーはテレマティクスを用いるシステムの発展とともに新車に占める台数が年々増加しており、2030 年には新規に出荷される乗用車のうち 95%以上を占めると予想されている [?]。

2.3 車両データを用いるサービス

実際にコネクティッドカーから抽出したデータを利活用するサービスは多く存在するが、本節ではその中でも特に運転歴に関する車両データを用いるサービスについて紹介する。運転歴に関する、とは速度や加速度、位置情報や時間、あるいはスロットル開度やステアリング角度など、自動車の動きの中でドライバーの「操作する」という動きと直接繋がって変化するデータであるところでは定義する。なお、自動車からデータを抽出する方法としては OBD-II コネクタやスマートフォン、EDR (Event Data Recorder。車載ブラックボックスとも呼ばれる) などが存在する。

2.3.1 利用ベース保険 (UBI)

利用ベース保険 (UBI) はテレマティクス保険とも呼ばれ、自動車から収集した運転データを基に保険料を算出する保険である。従来の自動車保険は大量の法則に基づき、過去の膨大な統計データを元に保険料を算出していたが、利用ベース保険ではより詳細な、

運転手ごとにカスタマイズされた保険料を算出することができる。例えば、責任感を持ち、安全な運転志向を持つドライバーは大きな割引を受けることができる一方、危険な運転志向を持つドライバーに対しては保険料を高く設定することができ、これにより保険会社のリスクを低減することができる。なお、保険料の算出方法として主流なものはPAYD (Pay As You Drive) と PHYD (Pay How You Drive) であり、前者は走行距離に連動し、後者は運転の仕方に連動するものである。運転の仕方には、平均速度、加速や減速の度合い、運転する場所、運転する時間などが含まれる。利用ベース保険は特に北米市場で高い需要があり、2030 年までには世界市場で 11.34% の年平均成長率で拡大すると予測されている [?] [?]]。

2.3.2 カーシェアリングサービス

カーシェアリングサービスは、登録された自動車を会員が共同で使用するサービスである。登録される自動車の所有者が個人であるか法人であるかや、登録される自動車の種類によって異なる様々なサービスが存在する。組織的なカーシェアリングサービス自体は 1980 年代後半にヨーロッパで始まり、日本では 2002 年に初の民営会社が発足したように決して新しいものではないが、近年のシェアリングエコノミー及びコネクティッドカーの普及により事業者数・利用者数ともに急激に規模を拡大しているサービスでもある。現在のカーシェアリングサービスでは、利用者が利用開始時に自身のスマートフォン内のモバイルアプリケーションを用いて自動車を解錠したり、利用者の運転歴データを事業者が収集したりすることが一般的である。例えば、パーク 24 グループが運営するカーシェアリングサービスであるタイムズカーにおいては、急加速・急減速などをリアルタイムで観測し、これらのデータを元に安全な運転を判定したうえで利用者にポイントを付与している。無事故走行距離とポイントに応じて利用者は 4 段階に分かれたステージに振り分けられ、ステージに応じて割引などの恩恵を受けられる。

2.4 本章のまとめ

本章では、コネクティッドカーやテレマティクス、車両データを用いたサービスなど、本研究の背景について紹介した。次章では、本研究の基盤となる技術について紹介する。

第3章 基盤技術

本章では、提案するシステムの基盤となる技術について紹介する。

3.1 デジタル証明書と構成技術

本節では、本研究におけるデジタル証明書の定義と、その構成技術について解説する。

3.1.1 デジタル証明書

通常、デジタル証明書は公開鍵と公開鍵の所有者を紐づけ、公開鍵の真正性を示す証明書を指すことが一般的だが、本研究では、ある主体に対する証明書発行者の主張を表現したデジタルデータと定義する。これらの主張には身分証明に関する個人情報から資格情報、ひいては大学の成績など多様な情報が含むことができ、デジタル証明書はこれらの情報を電子的に表現した上で第三者に対する提示・検証を可能とするための手段である。一般的に、デジタル証明書は以下のような要素を持つ。

- 発行者情報：デジタル証明書の発行者を識別するための情報
- 証明書の主体：証明書の主張の対象となる主体を識別するための情報
- 主張内容：発行者が証明書の主体について主張するなんらかの情報
- 有効期限：証明書の有効期限に関する情報
- 改ざんされていないことの証明：証明書が改ざんされていないことを示すための情報

これらの要素はデジタル証明書のデータモデルに従い表現され、様々な標準化団体が多様なデータモデルを提案している。その一例である、W3C による Verifiable Credentials Data Model については後述する。また、デジタル証明書はシリアライズされてデジタルデータに変換されるが、代表的なシリアライズ形式としては JSON, CBOR などがある [?][?]。

3.1.2 暗号学的ハッシュ関数

ハッシュ関数とは、任意の長さのデータから要約された値（ハッシュ値）を得る関数であり、特に暗号や情報セキュリティの用途に適した性質を持つものを暗号学的ハッシュ関数と呼ぶ。同じ入力値からは常に同じハッシュ値を得られ、またハッシュ値から入力値を計算することは困難であるという性質がある。後述する公開鍵暗号やデジタル署名でも用いられている技術であり、以下の特性を持つ。

- 原像計算困難性：ハッシュ値から元の入力値を計算することが困難である
- 弱衝突耐性：ある入力値とハッシュ値のペアがあったとき、同じハッシュ値を持つ異なる入力値を計算することが困難である
- 強衝突耐性：同じハッシュ値を持つ 2 つの異なる入力値のペアを計算することが困難である

なお代表的な暗号学的ハッシュ関数としては、SHA-256 などが挙げられる。

また、ハッシュ値を用いた技術としてハッシュチェーンが存在する。ハッシュチェーンは、連続するハッシュ値を順番に連結して作られるチェーン状のデータ構造であり、含まれるデータの整合性を保ち、改ざんの検知が可能である。ブロックチェーンにおいて、個々のブロックの整合性を保証するために使用されているのも、ハッシュチェーンである。

3.1.3 公開鍵暗号

公開鍵暗号は、暗号化と復号で異なる 2 つの鍵を使用する暗号方式である。2 つの鍵はそれぞれ秘密鍵と公開鍵と呼ばれ、秘密鍵はその所有者によって秘密を保って管理される鍵であり、秘密鍵と対になる公開鍵は広く公開され、他人が利用可能な鍵である。公開鍵暗号は広く用いられており、例えばインターネット上の通信を保護するためのプロトコルである SSL/TLS で利用されている。公開鍵暗号の代表的なアルゴリズムとしては、RSA(Rivest-Shamir-Adleman)、DSA(Digital Signature Algorithm)、ECDSA(Elliptic Curve Digital Signature Algorithm) などが挙げられる [?]。

3.1.4 デジタル署名

デジタル署名は、あるデジタルデータが特定の主体により作成されたこと、かつ作成されたのちに改ざんされていないことを検証可能にするための技術である。デジタル署名は、以下のプロセスで作成・検証される。

- 作成プロセス
 1. ハッシュ値の計算：署名者は、署名したいデータからハッシュ値を計算する

2. 秘密鍵による署名：計算されたハッシュ値に対して、署名者の秘密鍵を使用して署名を生成する。この生成された値がデジタル署名となる

- 検証プロセス

1. 公開鍵の入手：まず、検証者は署名者の公開鍵を入手する。通常、公開鍵は、公開鍵と公開鍵の所有者を紐づける証明書の形で配布されている
2. ハッシュ値の再計算：受け取ったデータをハッシュ関数に通し、ハッシュ値を再計算する
3. 署名の検証：署名者の公開鍵を用いて複合したハッシュ値と再計算したハッシュ値を比較し、同じであれば検証が成功したと言える

3.2 Verifiable Credentials と Decentralized Identifiers

本節では、デジタル証明書の一形態である Verifiable Credentials[?] と、それを支える識別子である Decentralized Identifiers[?] について解説した後、デジタル証明書の文脈で重要な機能である選択的開示と、アプリケーションとして欠かせないウォレットアプリケーションについても解説する。

3.2.1 Verifiable Credentials

前節で解説したデジタル証明書の一つの形態として、Verifiable Credentials がある。Verifiable Credentials は W3C が標準化を進めており、一般的に Verifiable Credentials といえば W3C による “W3C VC” を指す。Verifiable Credentials はデジタル証明書そのものであり、Verifiable Credentials のためのデータモデルとして Verifiable Credentials Data Model が存在する。広義の Verifiable Credentials として、ISO/IEC 18013-5 で定義された mdoc/mDL[?] や、IETF が標準化を進める SD-JWT VC[?] などにも存在するが本稿では詳しくは触れない。なお、Verifiable Credentials は “VCs”、Verifiable Credentials Data Model は “VCDM” と呼ばれることが一般的なため、本稿でもこれ以降この呼称を用いる。

VCs とは、検証可能な証明書のことであり、暗号学的手法で改ざん検知、および正当に発行されたものであるかの検証を可能にしたクレデンシャルのことである。クレデンシャルとは、発行者によって発行された、対象となる主体に対する何らかの主張（クレームと呼ばれる）の組と定義される。例えば、髪が黒く、眼鏡をかけた男性がいた場合、クレームは「髪が黒い」「眼鏡をかけている」「男性である」の3つとなり、これをまとめたものがクレデンシャルとなる。さらにこのクレデンシャルを暗号学的に改ざん検知、正当に発行されたものであるか検証できるようにしたものが VCs、といったイメージである。

VCs の基本的な構造は、発行者 (Issuer)、保持者 (Holder)、検証者 (Verifier) という三者の役割分担を前提としている。なお、この三者が登場するモデルは IHV モデルと呼ばれる。VCs の発行から検証までの一連の流れは下記の通りである。

1. 発行：発行者が特定の主体（Subject と呼ばれる）に関するクレデンシャルを用意し、暗号学的な署名を施す。完成したら、Holder に送信する。多くの場合 Subject と Holder は一致するが、異なっても構わない
2. 保持・提示：Issuer から送信された VC's を受け取り、後述するウォレットアプリケーションなどに保持する。任意のタイミングで Verifier に VC's を提示する
3. 検証：Verifier は Holder から送信された VC's から検証に必要な公開鍵情報を取り出し、公開鍵を用意した上で VC's を検証する

また、各主体はデータを Verifiable Data Registry と呼ばれるデータ保管場所とやり取りする。一連の役割分担を示したのが下図である。

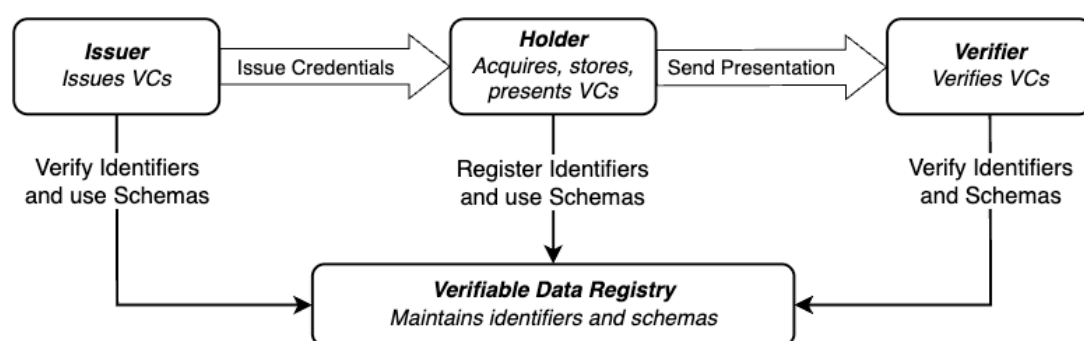


図 3.1: VC's における IHV モデルと仕組み

また、Holder が複数の VC's を保持し、場合によってはまとめて Verifier に提示するユースケースがあることは容易に考えられる。そのため、VC's には複数の VC's を合わせて提示する Verifiable Presentations という形式も存在する。Verifiable Presentations には、その Verifiable Presentations が正しく構成されていることを示す証明と、提示者が Holder であることを示す証明が含まれる。なお、Verifiable Presentations も “VP” と一般的に呼ばれるため、本稿でもこれ以降この呼称を用いる。

3.2.2 Decentralized Identifiers

Decentralized Identifiers は識別子の一種であり、ブロックチェーンなどの分散型台帳技術と紐づけられる。従来のアイデンティティ管理システムで用いられるような ID とは異なり、中央集権的な構造を持たないため異なるシステム間で相互運用性を持たせられたり、ID を自己管理できたりするというメリットがある。先述した VC's でも各主体の識別子としては Decentralized Identifiers が用いられることがほとんどである。なお、Decentralized Identifiers は一般的には “DIDs” と呼ばれるため、本稿でもこれ以降この呼称を用いる。

DIDs 自体は下図のような構造を持つ。“Scheme” はこの識別子が DIDs であることを示

す。“DID Method”はこのDIDsがどのタイプのDIDsであるかを示すもので、主にDID Document（後述）の置き場所により多くの種類が存在し、代表的なものではdid:web, did:key, did:ionなどが挙げられる[?] [?] [?]。 “DID Method-Specific Identifier”はDID メソッドの名前空間内での一意な識別子である。



図 3.2: DIDs の例

また、DIDs は下図のようなエコシステムを持つ。DIDs は単体では識別子に過ぎないため、DIDs に関するより詳細な情報を記載したデータとして “DID Document” が存在し、さらに DID Document を補助する役割を果たす “DID Controller” と “DID URL” が存在する。“Verifiable Data Registry” は DIDs, DID Document の置き場所として利用され、ブロックチェーンなどの分散型台帳技術が用いられることが多い。なお、DIDs から DID document を引くことを “resolve” といい、本稿ではこれ以降**解決**と呼称する。

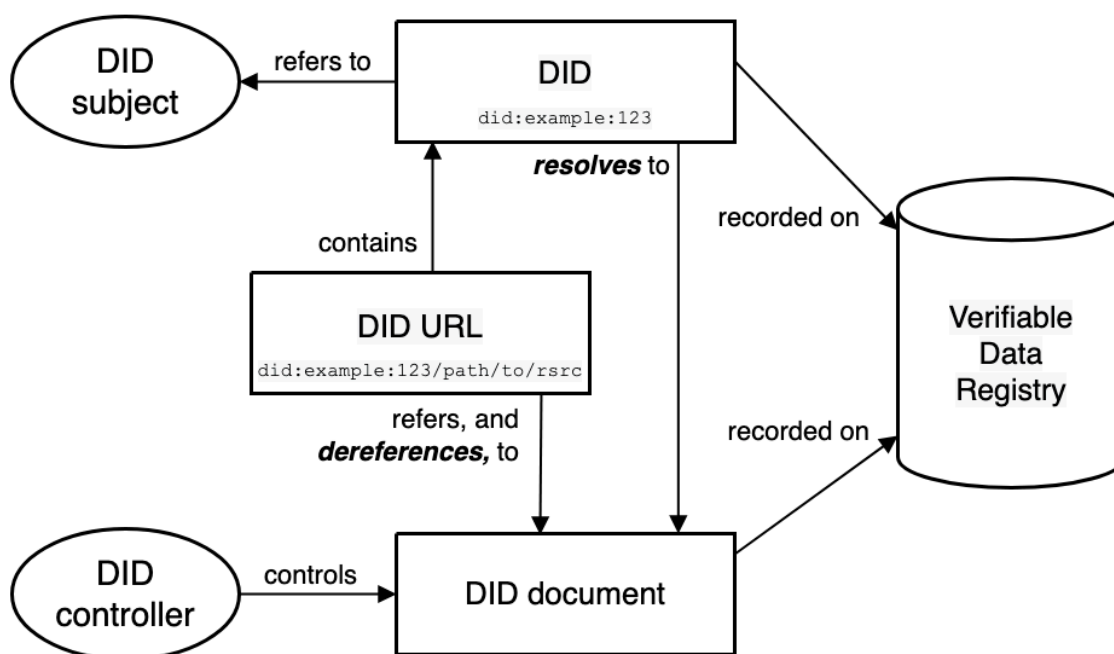


図 3.3: DIDs のエコシステム

3.2.3 選択的開示

VCs や DIDs の文脈で重要な機能の一つが、選択的開示である [?]。選択的開示とは、Holder が Verifier に証明書を提示する際、開示する内容をコントロールできる機能である。例えば、20 歳以上であることを示しアルコール飲料を購入するユースケースでは、従来ならば運転免許証などを全て開示していたが、選択的開示を用いることで顔写真と年齢の欄のみを開示すればよくなる。選択的開示を用いることで、Holder は Verifier に対して必要以上の情報を開示することがなくなり、Holder のプライバシーを保護することにつながる。選択的開示の具体的なアルゴリズムとしては、Selective Disclosure for JWTs(SD-JWT)[?] や、ゼロ知識証明を用いたものなどが提案されている。

3.2.4 ウォレットアプリケーション

IHV モデルにおいて、Holder が VCs などのデジタル証明書を保持するには、何らかの器が必要である。そこで登場するのがウォレットアプリケーションであり、Holder が複数の Issuer から発行された多様なデジタル証明書を格納し、自在に組み合わせて Verifier に提示するのに用いられる。

3.3 本章のまとめ

本章では、基盤技術としてデジタル証明書とその構成技術について紹介した上で、デジタル証明書の一形態である VCs とそれに関連した一連の技術についても紹介した。次章では、既存システムにおける問題点について議論する。

第4章 本研究における問題定義

本章では、既存システムにおいて本研究が着目する問題について述べる。

4.1 既存システムにおけるデータの流れ

2.2 節及び 2.3 節で紹介した既存サービスにおいて用いられているシステムでは、基本的にコネクティッドカーから収集されたデータは自動車からサービスの事業者が接続しているクラウドへと流れ、クラウドで処理をされた後にサービス事業者・メーカー事業者へと提供されている。システムのデータの流れを見てみると、基本的にデータは自動車と事業者間を繋ぐネットワーク内にのみ存在していると言える。トヨタの提供する T-Connect のように、モバイルアプリケーションへの配信などを通じて利用者にデータを提供するサービスも一部存在するが、あくまでサービスに一環としてわざわざデータをモバイルアプリケーションにも提供しているに過ぎず、本質的なデータの流れは変わらないと言える。

4.2 既存システムの問題点

前節で述べたデータの流れは、既存サービスのような登場人物が利用者と事業者のみ、というような一対一のユースケースにおいては十分である。しかしながら、特定の事業者とのみデータを共有する既存システムは、ベンダーロックインを容易に招き、利用者にサービスの乗り換えという選択肢を与えない。また、データを共有と言えど、実際のデータは利用者の自動車から事業者のクラウドに蓄積されているため、利用者が自身の自動車から抽出したデータへのアクセスが可能であるとも言い難い。例えば、利用者が保険会社を乗り換えても利用ベース保険の実績を引き継ぐことができるかどうかや、カーシェアリングサービスをまたいでも安全運転の履歴により割引を受けられるかどうかはひとえに保険会社やカーシェアリングサービスの判断に委ねられている。実際に例で言うならば、タイムズカーにおいては、無事故走行距離とポイントに応じて利用者は4段階に分かれたステージに振り分けられ、ステージに応じて割引などの恩恵を受けられるが、このステージは他のカーシェアリング事業者（オリックスカーシェアや三井のカーシェア）においては全く意味をなさない。言い換えるならば、複数の事業者にまたがってデータを共有する際、既存サービスでは事業者やメーカー間の連携が不可欠であるという問題点が存在する。また、このようなシステムでは、複数の事業者が結託して利用者に断りなくデータを共有する事態が発生する可能性も否定できない。

次項からは、この問題を構成している要因についてそれぞれ述べる。



図 4.1: 既存システムにおけるデータの流れ

4.2.1 データのサイロ化

前述したように、既存サービスを構成しているシステムにおいてはコネクティッドカーから収集されたデータは自動車から事業者が接続しているクラウドへと流れ、クラウドで処理をされた後に事業者へと提供される。このようなシステムでは、自動車と事業者の共有するネットワークは非常に閉鎖的であり、常にその間にデータが閉じているためサイロ化とでも言うべき状態が発生する。利用者が複数事業者を利用する場合、互いに独立したサイロが事業者の数だけ複数存在することになり、これが複数の事業者間の連携を難しくしている要因の一つである。

4.2.2 データのコントロール権の欠如

また、サイロが構成されていると言っても、データはコネクティッドカーから抽出された後に事業者と接続したクラウドにて処理・蓄積されるため、利用者が実際にデータにアクセスしたり、データをコントロールしたりすることは不可能に近い。さらに言うならば、利用者は自身の自動車からどのようなデータが事業者に渡り、それがどのように扱われているかについて知ることはできない。データが利用者のアクセス・コントロール可能な場所にないため、事業者が仮にデータを不正に利用したり、転売したとしても利用者がそれを検知することができないという問題点も存在する。仮に利用者が自分の自動車から抽出されたデータにアクセスし、それをコントロールすることが可能であれば、そのデータを複数の事業者に対して提示できる可能性が出てくるが、そのようなシステムは現在存在しない。

4.2.3 データの真正性の担保

仮に利用者のアクセス・コントロール可能な場所にデータが移り、その上で利用者がデータを提示してきたとしても、それを事業者が真正であると検証できるとは限らない。自動車から直にデータを抽出するアーキテクチャでない以上、利用者がデータを偽造・改ざんしていたり、他人の自動車から抽出したデータを提示してきたりする可能性を否定で

きないからである。そのため、既存システムではデータの真正性を担保するために、事業者が認めた自動車メーカーが製造したコネクティッドカーから安全な通信経路を経由してクラウドにデータを送信しているとも言える。

4.2.4 各事業者のポリシーの相違

さらに、これまでに述べた要因が取り除かれたとしても、各事業者のポリシーの相違によってデータを共有することができない場合がある。これには2つのパターンがあり、データ取得時のポリシーと、データ利用時のポリシーの差異がある。データ取得時のポリシーとは文字通りどのようなデータを取得するかを判断する際のポリシーであり、例えばA社ではスロットル開度を取得しない一方B社では取得しているとしたら、A社からB社へデータを持ち越すのは困難になる。データ利用時のポリシーとは取得したデータを用いて利用者を判断する際のポリシーであり、先ほどの例ではタイムズカーではこれに基づき利用者を4段階のステージに振り分けるが、ポリシーを共有しない他社ではこのステージでは全く意味がなくなってしまう。このような場合はポリシー実行後の状態のみを共有するのではなく、元のデータとポリシーの両方を合わせて提示するなどの工夫が必要になるだろう。いずれにせよ、各事業者はそれぞれ異なるポリシーを持っており、判断に用いられるデータは同じでも、例えば保険会社では総合した安全運転の実績を重視する一方、カーシェアリングサービスでは急な加減速の有無を重視するなど、ポリシーによってデータの評価基準が異なってしまう、複数の事業者にまたがってデータを共有することを難しくしている。

4.3 本章のまとめ

本章では、既存サービスを構成しているサービスにおけるデータの流れを示し、その問題点を洗い出した。次章では、以上の問題点を解決するシステムを提案する。

第5章 提案手法

本章では、第4章で提示した問題を解決するために、5.5節で提示する前提条件のもとで利用者のデータへのコントロール可能性とデータの真正性の担保を両立する手法およびシステムを提案する。

5.1 提案手法の概要

第4章で示したように、既存システムではデータが自動車と事業者の間に閉じることでデータがサイロ化し、利用者にとってサービスの乗り換えが難しくなり、また自身の自動車から抽出したデータにアクセス・コントロールすることができないためデータがどのように扱われているか感知できないという問題が存在する。また、事業者から見てもポリシーの差異によりデータを元に判定・スコアリングされた利用者をそのまま受け入れることはできず、さらにデータそのものを提示しても自動車から直に抽出した物でない以上利用者による改ざん・なりすましのリスクがつきまとうため受け入れることができないことも示した。

以上の問題を解決するために、提案手法ではデータの流れに新たに利用者を加え、自動車から抽出したデータをVCsで表現した上で利用者に渡す。利用者はVCsを受け取った後、ウォレットアプリケーションに格納し、望むタイミングでVCsないしはVPとして検証者に提示することができる。検証者はVCsないしはVPに含まれる公開鍵情報をもとに検証を行い、データの真正性を検証できる。VCsの特性により、検証者は当該VCsが適切な自動車から発行された物であることが確認でき、利用者も自身の自動車から抽出したデータに対するアクセスおよびコントロールを獲得できる可能性がある。本研究ではこの手法が実現可能であることを示すために、提案手法に沿ったシステムを設計した。

5.2 全体構造

本節では、まず提案するシステムのアーキテクチャについて示した後、データおよび処理がどのように流れるかを示し、さらに具体的なコンポーネントとしての自動車、モバイル、検証者のそれぞれが持つアプリケーションについて説明する。

5.2.1 システムアーキテクチャ

提案するシステムの論理アーキテクチャ図は以下の通りである。

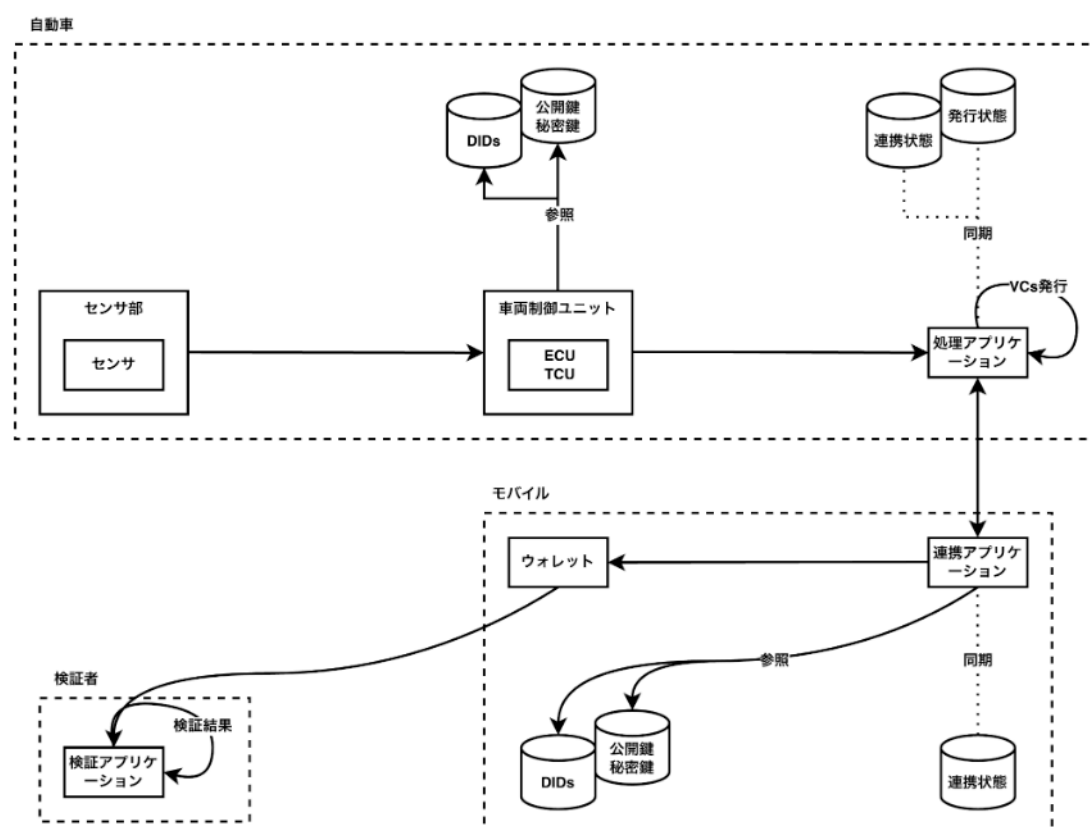


図 5.1: 提案するシステムの論理アーキテクチャ

本システムは上図が示すように、自動車、モバイル、検証者の3つのコンポーネントから構成される。

自動車は通常コネクティッドカーが持つ機能・コンポーネント（各種センサ、ECU や TCU などの車両制御ユニット、鍵ペア、識別子など）に加え、データを処理するための処理アプリケーションを持つ。処理アプリケーションの持つ機能は、モバイルとの連携、自動車から取得したデータを元に VCs を発行、VCs の送信、状態管理である。モバイルは通常モバイルが持つ機能・コンポーネント（鍵ペア、識別子など）に加え、自動車と連携するための連携アプリケーションと VCs を保持・提示するためのウォレットアプリケーションを持つ。連携アプリケーションの持つ機能は自動車との連携、VCs の受け取り、ウォレットアプリケーションへの送信、状態管理であり、ウォレットアプリケーションの持つ機能は VCs の保持、VP の構成、VCs および VP の提示である。検証者は検証アプリケーションを持つ複数の事業者である。検証アプリケーションは提示された VP から公開鍵を解決し、真正性の検証を行う機能を持つ。以下は、各コンポーネントが持つアプリケーションとその機能をまとめた表である。

表 5.1: システムの各コンポーネントが持つアプリケーションとその機能

コンポーネント	アプリケーション	機能
自動車	処理アプリケーション	モバイルとの連携、VCs の発行、VCs の送信、状態管理
モバイル	連携アプリケーション	自動車との連携、VCs の受け取り、ウォレットアプリケーションへの送信、状態管理
	ウォレットアプリケーション	VCs の保持、VP の構成、VCs および VP の提示
検証者	検証アプリケーション	公開鍵の解決、真正性の検証

5.2.2 データおよび処理の流れ

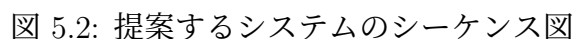
提案するシステムのシーケンス図は以下の通りである。システムは、処理の段階に応じて 3 つのフェーズに分類される。それぞれ連携フェーズ、データ処理フェーズ、検証フェーズであり、関係するコンポーネント・アプリケーションが異なる。

まず、連携フェーズにおいて、VCs を発行するべく自動車の処理アプリケーションとモバイルの連携アプリケーションが連携する。連携の際、処理アプリケーションは自動車内の、連携アプリケーションはモバイル内の鍵ペアをそれぞれ用い、チャンレンジレスポンス認証を行った上で通信経路を確立する。通信経路が確立すると、それぞれのコンポーネント内で連携の状態を管理しているデータベースなどと通信し、連携状態を同期する。

自動車とモバイルの連携が完了すると、実際に VCs を発行することができるため、処理フェーズへ移行する。自動車が各種センサからデータを取得し、EUC や TCU などの車両制御ユニットを介して処理アプリケーションへとデータが送られる。処理アプリケーションでは同期されている VCs 発行状態を参照しながら、連携時に自動車から取得した鍵ペアを用いて VCs を作成し、連携フェーズで確立した通信路を用いてモバイルに VCs を逐次送信する。モバイルでは連携アプリケーションが VCs を逐次受け取り、ウォレットアプリケーションへと送信する。ウォレットアプリケーションでは受け取った VCs を格納し、保持する。

検証者が検証情報の提示を求める状態が発生すると検証フェーズへと移行する。このような事態になる要因としては、利用者が新規サービスを申し込む、既存サービスを乗り換える、などが考えられる。利用者はモバイル内の鍵ペアを用いてウォレットアプリケーションに格納された VCs から VP を構成し、検証者へと提示する。なお、実際のデータは複数回にわたり VCs へ処理されることが想定され、また車両の DIDs の真正性を担保するための VCs も必要であることから利用者が検証者へ提示するのは必然的に VP になる。検証アプリケーションは VP を受け取ったら、含まれる DIDs から DID Document を解決し、自動車と利用者の公開鍵情報を取得する。取得した公開鍵情報を元に VP を検証し、検証結果を利用者へと返す。

以上が一連の処理の流れである。



処理アプリケーションは、モバイルの連携アプリケーションとの間の通信経路の確立と、自動車内の各種センサから取得した情報をVCsに加工し、連携アプリケーションに送信するという2つの機能を持つ。通信経路の確立は、公開鍵基盤に基づいて行われ、自動車とモバイルアプリケーションそれぞれが持つ鍵ペアを用いて行われる。また、自動車はメーカーから割り振られた一意の自動車DIDsと、VCsに添付するための複数のデータDIDsを持つ。VCsを発行する際に、自動車DIDsとデータDIDsが紐づくことを示すVCsを添付することにより、データDIDsの出所を担保する。なお、この紐付きを示すVCsは、連携時に一度だけ添付すれば十分である。また、データをVCsに加工するとき用いる鍵ペアは、連携アプリケーションとの間の通信経路を確立するとき用いたものと同じものを用いる。

5.2.4 連携アプリケーション

連携アプリケーションは、自動車の処理アプリケーションとの間の通信経路の確立と、処理アプリケーションから送信された VCs をウォレットアプリケーションに送信するという 2 つの機能を持つ。なお、通信経路の確立については、前項で述べたとおりである。連携アプリケーションは、連携時にモバイルの鍵ペアを用いる。

5.2.5 ウォレットアプリケーション

ウォレットアプリケーションは、同じモバイル内の連携アプリケーションから送信された VCs を格納・保持することと、VCs から VP を構成した上で検証者に提示するという 2 つの機能を持つ。ウォレットアプリケーションは、VP の構成時にモバイルの鍵ペアを用いる。

5.2.6 検証者アプリケーション

検証者アプリケーションは、実際の事業者により多様に異なる実装をとると想定されるが、基本的な役割としては、送信された VP を確認してその有効性及び真正性を検証することである。すなわち、必要な機能としては受け取った VP をパースした後含まれる DIDs から DID Document を解決し、自動車および利用者の公開鍵情報を取得して VP が適格に発行され、改ざんされておらずかつ失効していないことを検証することである。

5.3 データモデル

システム内で各コンポーネントとアプリケーションが連携するには、連携の基盤としてのデータモデルが必要である。本節では、具体的にどのようなデータがデータモデル含まれ、使用されるかについて述べる。

5.3.1 基本設計

基本設計として、データモデルには抽象化された自動車情報が含まれ、一定時間の間の状態を記述する。前提として、自動車から取得される情報は、車両やセンサごとに種類や精度にばらつきがあり、それら全てを網羅したデータ構成を用意することは非常に難しい。また、車両の性質による特殊な情報やメーカーのポリシーに依存した情報もあり、これらも全て含むことは非常に困難であるばかりかデータ量を増大させ、システムの肥大化をも招いてしまう。そのため、検証に必要であると想定されるデータセットを定義することでこれらのばらつきや差を吸収し、システムの基盤として利用していく必要がある。

なお、データは正規化されていることを想定している。例えば、センサのパルス数ではなく速度として算出されたものを使用する、電圧ではなく on/off の状態として算出されたものを使用するなどである。

5.3.2 データの構造

データセットとして、本研究では速度、加速度、回転角速度、ステアリング角度をデータモデルに含むこととした。各データは全て自動車の状態である。データモデルの策定に際しては、PHYD タイプの UBI 保険や道路交通安全に関連する国際規格・国際標準を参照した [?] [?] [?] [?]]。以下は、データモデルに含まれるデータが何を検出するためか、および実際にどのようなセンサから取得できるかを示した表である。

表 5.2: システムで使用するデータとセンサ

データ	目的	センサ
速度	速度超過率の検知	車速センサ
加速度	急加速・急減速および加速度変化率の検知	加速度センサ
回転角速度	急旋回の検知	ジャイロセンサ
ステアリング角度	急旋回の検知	舵角センサ

また、以下は各データについての簡単な説明である。

速度 速度はより正確には車速と呼称され、文字どおり車の速さを表すベクトル値である。道路を設計するときに対象とされる自動車の速度（設計速度）などに基づき道路には最低速度・最高速度が規定され、道路の利用者は規定に従わなければならない。なお、規定に反した場合は刑事処分の対象となる。単位としては km/h が用いられる。

加速度 加速度とは決められた時間内で速度に変化が生じる割合のことである。加速度の変化を計測することによって、速度のみでは知り得なかった自動車の状態の変化、例えば急加速や急減速を検知することができる。単位としては m/s^2 が用いられる。

回転角速度 回転角速度とは、回転運動時の回転速度である。回転角速度を計測することで、自動車の旋回の状態、例えばコーナーをどのように曲がったかを検知できる。単位としては rad/s が用いられる。

ステアリング角度 ステアリング角度とは舵角とも呼称され、ステアリング（ハンドル）が方向転換時に切れる角度、すなわち切角のことである。ステアリングの最大切角を回転数で示したものがロックトゥロックであり、実際の値は自動車ごとにギア比などによって異なるが、一般的な乗用車がだいたい 1080° 900° （おおよそ 3 回転 2.5 回転）である。単位としては $^\circ$ が用いられる。

5.4 システムの満たすべき要件

提案するシステムは、以下の要件を満たすことが望ましい。

5.4.1 データのコントロール可能性

利用者が自身の自動車から抽出したデータにアクセスし、コントロールできること。ここで言うコントロールとは、データがどこに所在しており、かつどのような内容が含まれるか利用者自身が把握しており、また任意の事業者に対してデータを提示することが可能なことを指すものであり、利用者が自由にデータを改変したり移転したりすることができることを指すものではない。

5.4.2 データの真正性の担保

データを検証した者は誰でも、そのデータが偽造・改ざんされていないことを確認できること。具体的には、検証者が利用者から提示されたデータを検証したとき、データが適格かつ望ましい自動車から取得されたものであり、利用者が他者になりすまして他者の自動車から取得したデータを提示してきていないことや、データを偽造あるいは改ざんした上で提示してきていないことを検証できることを指す。

5.5 前提条件

本提案手法は以下の前提条件を持つ。

自動車は純正である 自動車はメーカーが製造したままの状態であり、改造などを施されていない状態（純正）であること。例えば、自動車愛好家の間で行われるようなカスタム・チューニングを施された自動車などであれば、この条件を満たさない。ここで言う「改造」とは、メーカーが自動車を設計した段階で利用者が自動車はこのような使用すると想定した方法から著しく逸脱して使用されていないことを指し、利用者が世間と同程度のレベルでの使用をしていれば満たされるであろう条件である。センサや ECU, TCU などが改変されることはデータが改ざんされることと同義と言え、前述した要件を満たせなくなるため、このような改造された自動車は本提案手法の適用の対象外とした。

自動車メーカーは不正を行わない 自動車メーカーが自動車の製造時に不正な行為を行わないこと。例えば、自動車メーカーが実馬力よりも低い数値を公表したりしていた場合、不正行為にあたりこの条件を満たさない。検証に際し、事業者が自動車から抽出したデータが真正性を持つと考えているのは自動車メーカーが不正を行わないと信じているためであることは前述のとおりであるが、自動車メーカー自体が不正な行為を行うとその信用が損なわれてしまう。そのため、このような事例は本提案手法の検討対象外とした。

自動車はコネクティッドカーである 提案手法の対象となる自動車はコネクティッドカーであること。電子制御を持たない自動車は ECU や TCU を持たず、結果当然として鍵ペアなども持てないため、モバイルとの間に通信経路を確立したり、データを送

信することはおろか、データを抽出することすら困難な場合が多い。その場合、提案手法が一切適用できなくなってしまうため、このような自動車は本提案手法の適用の対象外とした。

5.6 本章のまとめ

本章では、第 4 章で提示した問題を解決するための提案手法を示し、具体的なシステムの構成・データモデル・要件を設計した上で前提条件にも言及した。次章では、概念実証として実際に動作するシステムを実装する。

第6章 実装

本章では、第5章で提案したシステムの概念実証としての実装について述べる。

6.1 実装の概要

前章で提案した手法を実装し、実際に動作するシステムを構築する。実装するのは自動車アプリケーション、モバイルアプリケーション、検証者アプリケーションの3つであるが、自動車アプリケーションに関しては実際の自動車で動作する車載アプリケーションを開発することは大きな労力を要するため、車載アプリケーションをエミュレートする形で実装した。また、実際に自動車からデータを抽出するのではなく実際のデータを模したテストデータを利用した。なお、この実装は提案手法の概念実証として行い、提案するシステムが十分に実現可能であるかどうかを検証することを目的としているため、提案システムのコアの部分、すなわちデータの真正性の担保とコントロール可能性の両立を実現する役割を持つコンポーネントを主に実装している。

6.1.1 技術スタック

実装にあたっての技術スタックとしては、以下を用いた。

表 6.1: 使用した技術スタック

要素	要素名
言語	TypeScript
VCs/VP ライブラリ	@trustknots/vcknots
DID メソッド	did:web
ハッシュアルゴリズム	SHA-256

また、did:web に対応した web サイトとして、<https://did.eunos.tech/>を用いた。

6.1.2 データモデル

データモデルには、第5章で提案したように速度、加速度、回転角速度、ステアリング角度が含まれる。実際の型定義としては、以下のように定義している。

プログラム 6.1: データモデルの型定義

```
1 interface SensorData {  
2     timestamp: number;           // タイムスタンプ  
3     speed: number;               // km/h  
4     acceleration: number;       // m/s2  
5     yawRate: number;            // rad/s  
6     steeringAngle: number;      // 度 ( ° )  
7 }
```

6.2 各アプリケーションの実装

6.2.1 自動車アプリケーション

前述の通り、実際の車載アプリケーションを実装するのは大きな労力を要するため、車載アプリケーションをエミュレートする形で実装している。自動車アプリケーションは以下の機能を持つ。なお、それぞれのそれぞれのコードスニペットや型定義については、付録を参照されたい。

SHA-256 ハッシュチェーンの生成 今回の実装ではサンプルデータを用いるが、サンプルデータを 1 秒ごとにハッシュチェーンに連結する。前提として、VCs はログではなく、主張の束である。速度、加速度、回転角速度、ステアリング角度といった自動車から取得されるこれらのデータは膨大なログであり、そのまま生で VCs に含むのは計算量的にも設計的にも不適切である。しかしながら、ログ自体の完全性を担保する必要もあるため、今回はハッシュチェーンを採用した。

データの集約 VCs を作成するために、60 秒間のデータを 1 つのセグメントとして扱い、それぞれのデータにつき平均値、最大値、最小値を算出する。データの完全性はハッシュチェーンの始まりと終わりをセグメントに含めることで担保する。

VCs の発行 データを集約し、セグメントごとのデータの平均値、最大値、最小値を主張として VCs を発行する。なお、発行される VCs は、W3C Verifiable Credentials 標準に準拠した構造を持つ。

モバイルとの通信経路の確立や状態管理、VCs の失効リストの管理を行う機能は実装していない。なお、VCs の送信・受信に関しては、アプリケーション内の関数を用いることで可能となっている。

6.2.2 モバイルアプリケーション

モバイルアプリケーションは以下の機能を持つ。

VCs の受信と保存 VCs を受け取り、データとして保存する。今回の実装では、JSON ファイルを簡易的にデータの保存場所として扱っている。

VP の構成 Holder の DIDs を取得し、受け取った VCs をもとに VP を構成する。なお、VCs と同様、VP も W3C Verifiable Presentations 標準に準拠した構造を持つ。

VP の提示 構成した VP を検証者に提示する。

VP 構成の状態管理、検証者との間の通信経路の確立を行う機能は実装していない。また、選択的開示についても特に実装は行っていない。

6.2.3 検証者アプリケーション

検証者アプリケーションは、実際の事業者により多様な形を取り得るが、共通する以下の機能を持つ。

6.3 本章のまとめ

第7章 評価

本章では、提案システムに対する評価について述べる。

7.1 評価内容

まず、5.4 節で示した要件を提案システムが満たしているかどうかを定性評価する。次に、STRIDE を用いた脅威分析を行い、システムが実運用に耐えうるかどうかを評価する。

7.2 システムの定性評価

第5章で提案・設計したシステムが5.4 節で示した要件を満たしているかどうか、以下でそれぞれ評価する。

7.2.1 データのコントロール可能性

データのコントロール可能性とは、利用者が自身の自動車から抽出したデータにアクセスし、コントロールできることである。本システムでは、利用者の自動車から抽出されたデータは自動車から直接モバイルへと送信され、利用者の手元でウォレットに格納されて任意のタイミングで検証者に提示することができる。利用者はデータがどこに所在するか、そしてどのようなデータが含まれるのかを認知することができるうえ、VP の提示という形で自身の自動車から抽出されたデータに対するコントロール権を獲得している。そのため、データのコントロール可能性を満たしていると言える。

7.2.2 データの真正性の担保

データの真正性の担保とは、データを検証した者は誰でも、そのデータが偽造・改ざんされていないことを確認できることである。本システムでは、データは自動車で VCs へと加工され、対応する利用者のウォレットアプリケーションへ格納された後に VP として提示される。提示された VP に含まれる DIDs から DID document を解決し、公開鍵を引いて VP を確認することで、当該データが適格な自動車から利用者に渡され、検証者に提示されたことおよび偽造・改ざんされていないことを検証できる。そのため、データの真正性の担保も満たしていると言える。

7.3 STRIDE による脅威分析

次に、STRIDE による脅威分析を行う。まず、STRIDE について整理した後にシステムのデータフローを示し、それを元に脅威を洗い出した上で評価を行う。

7.3.1 STRIDE について

STRIDE とは、Microsoft 社が提唱した脅威分析のモデリング手法であり、システムに対する脅威を 6 つに分類する [?]。脅威分析モデルを構築し脅威のリスクを調べることで、セキュリティホールの検知などに役立てることができる。STRIDE が分類する脅威のカテゴリは以下の通りである。

表 7.1: STRIDE における脅威のカテゴリ

カテゴリ	説明
Spoofing (なりすまし)	第三者が正規の主体を装う
Tampering (改ざん)	データやシステムの偽造
Repudiation (否認)	主体による行為の否定
Information Disclosure (情報漏えい)	機密情報の不正な流出
Denial of Service (サービス妨害)	正当な利用者がシステムを利用できない
Elevation of Privilege (権限昇格)	不正に上位の権限が取得される

なお、STRIDE には、DFD (次の節で言及) に含まれる要素全てを対象として脅威を洗い出す STRIDE-per-Element と、DFD の信頼境界上の脅威を洗い出す STRIDE-per-Interaction が存在するが、本稿では STRIDE-per-Element を採用する。

7.3.2 システムの DFD

STRIDE における脅威分析においては、まず対象システムの DFD(Data Flow Diagram)、すなわちデータフロー図を作成する。DFD はシステムの構成要素や機能、そしてデータの流れを示した図であり、データストア、プロセス、外部の主体、データフローという 4 種類の要素から構成される。各要素は信頼できる要素ごとにグループ化され、信頼境界という境界線で区切られる。なお、信頼境界は DFD においては点線で表現される。本システムにおいては、信頼境界は 3 つに分かれ、それぞれ自動車、モバイル、検証者である。また、STRIDE においては、脅威カテゴリと DFD の各要素との対応関係が定義されており、これに基づき各要素に対して検討すべき脅威カテゴリが決定される。本システムの DFD は下図のようになる。

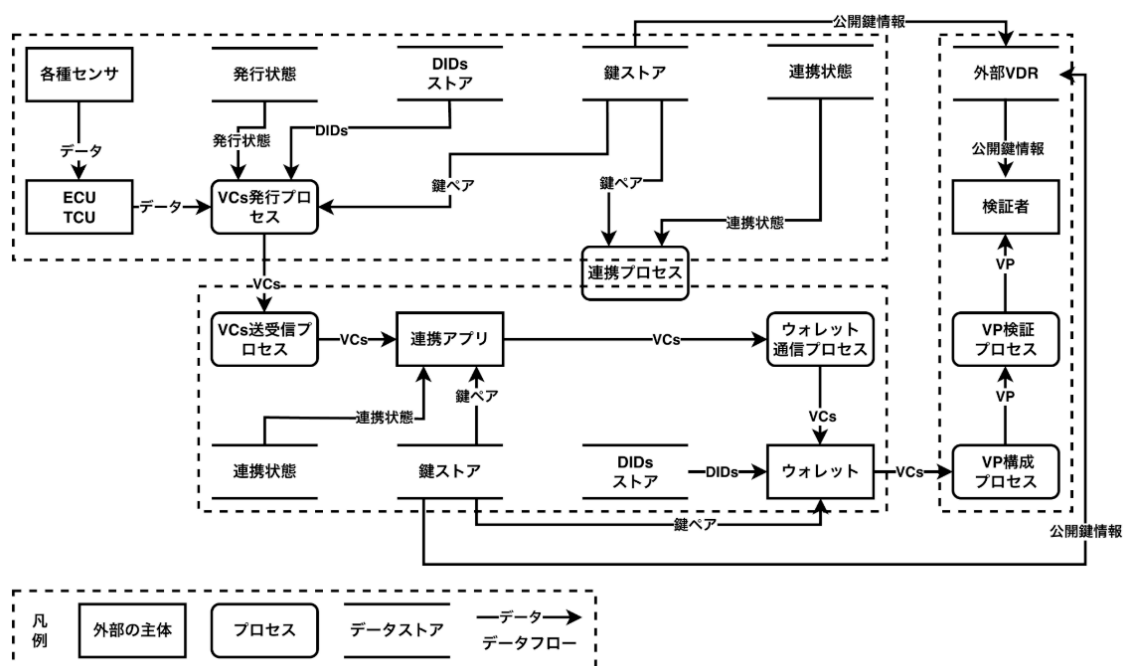


図 7.1: 提案するシステムのデータフロー図

7.3.3 特定された脅威

本システムでは、自動車、モバイル、検証者のそれぞれが信頼境界を形成している。DFDを元に、それぞれの信頼境界内の各要素について脅威分析を行った。脅威分析で特定された脅威は多くあるが、本研究の適用の範囲外のものも存在するため、より重要な脅威に絞って取り上げる。なお、脅威分析の結果の詳細は付録 A の「STRIDE による脅威分析の結果」に掲載したので参照されたい。

7.3.4 自動車における脅威

VCs 発行プロセス VCs 発行プロセスが改ざんされて不正に行われたり、正当な利用者になりすました第三者が VCs の発行を受けることを試みるような脅威である。この脅威により、VCs の正当性が意味をなさなくなったり、利用者を検証者が疑う必要などができてしまう。対策としては、VCs を発行するプロセスを耐タンパ性のあるセキュアな領域で実施する、利用者の認証を適格に行うなどが考えられる。なお、利用者の認証に関しては、超広帯域通信（UWB: Ultra-Wide Band）および低消費電力通信（BLE: Bluetooth Low Energy）を活用した研究などがなされている[?]。

連携プロセス 連携プロセスが改ざんされて不正に行われたり、正当な利用者になりすました第三者が連携をしようとしたりする脅威である。この脅威により、発行されたVCsが悪意ある第三者のもとに流れる可能性が生じ、さらに悪意ある第三者が正当

な利用者のふりをして検証者に VP を提示することなどにつながってしまう。対策としては、利用者の認証を適格に行うことが考えられる。

公開鍵情報 公開鍵情報が外部 VDR に格納されるまでの間に改ざんされる脅威である。この脅威により、検証者が VP を正当な手順で検証できないという問題が発生しうる。対策としては、安全が確立された通信経路を用いる、あるいは対処療法的に、検証者は VP を検証する前に取得した公開鍵の正当性を一回検証する、などが考えられる。

7.3.5 モバイルにおける脅威

VCs 送受信プロセス VCs 送受信プロセスが改ざんされて不正に行われたり、正当なウォレットアプリケーションになりすましたアプリケーションを操作する第三者が VCs の受信を受けたることを試みるような脅威である。この脅威により、正しく VP を構成できなくなるという問題が発生しうる。対策としては、連携アプリケーションとウォレットアプリケーションの連携を適格に行い、怪しい、認証していないようなアプリケーションと接続をしないようにすることなどが考えられる。

連携プロセス 自動車と同様の問題がモバイルでも発生しうる。

公開鍵情報 自動車と同様の問題がモバイルでも発生しうる。

7.3.6 検証者における脅威

検証者 外部の主体である検証者に、悪意ある第三者がなりすます脅威である。この脅威により、利用者と検証者が VP を提示・検証することで達成したい目的を達成できないという問題が発生しうる。対策としては、検証前に検証者の認証を利用者が適格に行えるような仕組みを作成することなどが考えられる。

7.4 本章のまとめ

本章では、提案するシステムに対し、5.4 節で示した要件を満たしているかの定性評価と、STRIDE による脅威分析という形で評価を行った。次章では、本研究のまとめと課題、そして展望を示す。

第8章 結論

本章では、本研究のまとめと今後の課題を示す。

8.1 本研究のまとめ

まとめます。

8.2 本研究の課題と限界

データ辞書の限界・事業者のポリシーに全て対応することはできないことを書きます。

謝辞

アリガトウ

付 録 A 付録

A.1 実装のコードスニペット

以下は実装のコードスニペットである。

プログラム A.1: ハッシュチェーンの生成アルゴリズム

```
1 hash_i = SHA256(  
2     prevHash (32 bytes) ||  
3     timestamp (8 bytes, big-endian double) ||  
4     speed (8 bytes, big-endian double) ||  
5     acceleration (8 bytes, big-endian double) ||  
6     yawRate (8 bytes, big-endian double) ||  
7     steeringAngle (8 bytes, big-endian double)  
8 )
```

プログラム A.2: セグメントの型定義

```
1 interface Segment {  
2     segmentStartTime: string;           // タイムスタンプ  
3     durationSec: number;  
4     aggregatedMetrics: {  
5         speed: { avg, max, min },  
6         acceleration: { avg, max, min },  
7         yawRate: { avg, max, min },  
8         steeringAngle: { avg, max, min }  
9     };  
10    hashChain: {  
11        start: string; // セグメントの最初のハッシュ  
12        end: string; // セグメントの最後のハッシュ  
13    };  
14 }
```

プログラム A.3: 発行される VCs の型定義

```
1 interface DrivingEvaluationCredential {  
2     id?: string;  
3     "@context": string[];
```



```

4     type: string[];
5     issuer: string;                      // 発行者のDIDs
6     issuanceDate: string;
7     credentialSubject: {
8         id?: string;                     // 被発行者のDIDs
9         segmentStartTime: string;
10        durationSec: number;
11        aggregatedMetrics: AggregatedMetrics;
12        hashChain: HashChain;
13    };
14    proof: {
15        type: string;
16        created: string;
17        verificationMethod: string;
18        proofPurpose: string;
19        jws?: string;
20        proofValue?: string;
21    };
22    }

```

プログラム A.4: 発行される VP の型定義

```

1  interface VerifiablePresentation {
2      "@context": string[];
3      type: string[];
4      holder: string;                      // HolderのDIDs
5      verifiableCredential: DrivingEvaluationCredential[];
6      // 含まれるVCs
7      proof: {
8          type: string;
9          created: string;
10         verificationMethod: string;
11         proofPurpose: string;
12         jws?: string;
13         proofValue?: string;
14     };
15 }

```

A.2 STRIDE による脅威分析の結果

第7章でSTRIDEによる脅威分析を行った。以下はその結果を信頼境界ごとに示した表である。なお、各脅威の番号は便宜上振り分けたものであり、別段の意味はない。

表 A.1: 自動車における脅威

番号	要素	要素の種類	脅威のカテゴリ	脅威の概要
1	VCs 発行状態	データストア	改ざん	VCs の発行状態が改ざんされる
2	DIDs ストア	データストア	改ざん	DIDs が改ざんされる
3	鍵ストア	データストア	改ざん	鍵情報が改ざんされる
4	連携状態	データストア	改ざん	モバイルとの連携状態が改ざんされる
5	VCs 発行プロセス	プロセス	なりすまし	第三者が VCs の発行を受ける
6	連携プロセス	プロセス	なりすまし	第三者が連携を試みってくる
7	VCs 発行プロセス	プロセス	改ざん	VCs の発行プロセスが不正に行われる
8	連携プロセス	プロセス	改ざん	モバイルとの連携プロセスが不正に行われる
9	DIDs	データフロー	改ざん	DIDs が改ざんされる
10	鍵ペア	データフロー	改ざん	鍵情報が改ざんされる
11	発行状態	データフロー	改ざん	発行状態が改竄される
12	連携状態	データフロー	改ざん	連携状態が改ざんされる
13	公開鍵情報	データフロー	改ざん	公開鍵情報が改ざんされる

表 A.2: モバイルにおける脅威

番号	要素	要素の種類	脅威のカテゴリ	脅威の概要
1	連携状態	データストア	改ざん	自動車との連携状態が改ざんされる
2	DIDs ストア	データストア	改ざん	DIDs が改ざんされる
3	鍵ストア	データストア	改ざん	鍵情報が改ざんされる
4	VCs 送受信プロセス	プロセス	なりすまし	第三者が VCs の受信を受ける
5	連携プロセス	プロセス	なりすまし	第三者が連携を試みってくる
6	VCs 発行プロセス	プロセス	改ざん	VCs の発行プロセスが不正に行われる
7	連携プロセス	プロセス	改ざん	モバイルとの連携プロセスが不正に行われる
8	連携アプリ	外部の主体	改ざん	連携アプリケーションが改ざんされる
9	ウォレット	外部の主体	改ざん	ウォレットアプリケーションが改ざんされる
10	DIDs	データフロー	改ざん	DIDs が改ざんされる
11	鍵ペア	データフロー	改ざん	鍵情報が改ざんされる
12	VCs	データフロー	改ざん	VCs が改竄される
13	連携状態	データフロー	改ざん	連携状態が改ざんされる
14	公開鍵情報	データフロー	改ざん	公開鍵情報が改ざんされる

表 A.3: 検証者における脅威

番号	要素	要素の種類	脅威のカテゴリ	脅威の概要
1	外部 VDR	データストア	改ざん	自動車や利用者の公開鍵情報が改ざんされる
2	検証者	外部の主体	なりすまし	第三者が検証者になりすます
3	鍵ストア	データストア	改ざん	鍵情報が改ざんされる
4	VP 構成プロセス	プロセス	改ざん	悪意ある形で VP を発行する
5	VP	データフロー	改ざん	VP が改竄される
6	公開鍵情報	データフロー	改ざん	公開鍵情報が改ざんされる

参考文献