博士論文 2025 年度

情報システムにおける信頼に基づくクレームの妥当性検証

慶應義塾大学大学院 政策・メディア研究科

阿部 涼介

博士論文要旨 2025 年度 (令和 7 年度)

情報システムにおける信頼に基づくクレームの妥当性検証

本研究は，主張者が主張するクレームの妥当性を検証者が判断する “クレームの妥当性検証” を情報システムで構成する抽象モデルを提案する．個人の実績証明や，多様化するシステムへのアクセスコントロール，フェイクニュースなどの偽情報に対して，デジタル証明書を活用した情報の妥当性検証が議論されている．デジタル証明書は様々な場面で繰り返し提示され，単体に限らず，複数の証明書を組み合わせて利用することも想定されている．しかし，証明書で示される情報であるクレームの妥当性検証は，既存のデジタル証明書の標準規格では検討の範囲外とされている．このため，典型的にはクレームの検証基準は特定のコンテキストに基づいて設計され，当該コンテキスト特有の仮定や条件が前提として組み込まれる．したがって，異なるコンテキストで特定のデジタル証明書を活用する場合，元々の前提を暗黙的に援用してしまい，目的に応じた適切な検証基準を構成できない恐れがある．そこで本研究では，デジタル証明書をはじめとする根拠情報を基に，検証者が自身のコンテキストに応じたクレームの妥当性検証を情報システムとして実現するモデルを提案する．まず，既存の証明書などを用いたやりとりをモデル化し，主張者が主張するクレームを，検証者が妥当とみなす検証基準を含む “検証ポリシ” への適合を判断する営みであると整理する．次に，検証ポリシに基づき，情報システムで処理するための関数モデルを定義する．ここで，形式論理に基づいて情報システムでクレームを検証する際，検証基準には計算機で決定困難な命題を含むことがある．そこで，検証基準を構成する一連の命題の中で，特定の命題が常に真であると仮定することを “信頼” であると整理する．その上で，信頼し，真と仮定する命題を明示しながら決定論的な検証基準を構成する “Shinken モデル” を提案する．Shinken モデルに基づき，信頼する命題を明示することで，検証基準の透明性と更改可能性を担保し，検証者のコンテキストに応じたクレームの妥当性検証を実現する．本研究では，Shinken モデルに基づいたクレームの妥当性検証の実証として，2 つのケーススタディを実施した．1 つ目は，過去の商取引における結果を検証するために，ブロックチェーンを改竄困難な台帳であると信頼し，仮定した上で検証基準を構成する取り組みである．2 つ目は，デジタル証明書を用いたクレームの妥当性検証において，証明書発行の根拠情報を証明書に紐付け，検証者が検証基準を更改する取り組みである．2 つのケーススタディから，明示的な信頼に基づき検証基準を構成し，情報システムとしてクレームの妥当性検証を実現可能であると評価した．本研究の成果に基づき，コンテキストに応じて信頼する対象を明示しながら，複数の根拠情報を活用し，情報の妥当性検証を情報システムとして実現することが期待される．

キーワード: 1. クレーム検証, 2. 信頼, 3. 妥当性, 4. デジタル証明書, 5. デジタル署名

慶應義塾大学大学院 政策・メディア研究科
阿部 涼介

Abstract of Dissertation - Academic Year 2025

Claim Validation based on Trust in Information Systems

This dissertation proposes an abstract model for "claim validation" in information systems, in which a validator judges the validity of claims asserted by claimants. Discussions are ongoing on validating information using digital certificates, such as validating individual achievements, access control for systems, and countermeasures against misinformation. In those discussions, it is expected that a claimant repeatedly presents the same certificate in various contexts and that the claimant presents multiple certificates in combination. However, the standards for digital certificates state that the validation of claims indicated in the certificates is out of the scope. Therefore, criteria for claim validation are typically designed based on a specific context incorporating assumptions and conditions unique to the context. As a result, when certificates are used outside their original context, the original assumptions may be implicitly adopted, making it difficult to establish validation criteria appropriate to the validator's purpose. Hence, in this dissertation, we discuss a model for claim validation in an information system, while utilizing evidence such as digital certificates. First, we organize the claim validation as a model of the interaction between the claimant and the validator. In our model, claim validation is a process by which a validator judges whether claims made by claimants comply with a "validation policy," which defines the validator's criteria for validity. Next, we discuss a function model for processing claim validation in an information system according to the validation policy. In practice, validation criteria may include propositions that are undecidable by a computer. To handle such cases, we define "trust" in claim validation as the assumption that a specific proposition is true within the set of propositions forming the validation criteria. We then propose the "Shinken model," which constructs deterministic criteria while explicitly specifying the propositions to be trusted. By explicitly stating the propositions to be trusted, the Shinken model ensures both transparency and adaptability of the validation criteria and realizes claim validation according to the validator's objectives. In this dissertation, we conducted two case studies of claim validation based on the Shinken model. The first is an attempt to construct validation criteria based on the Shinken model, trusting that the blockchain is a tamper-resistant ledger, to validate the result of a past commercial transaction. The second case study is an attempt to validate claims using digital certificates by linking the evidence for certificate issuance to the certificates and allowing the validator to modify the validation criteria. From the two case studies, we conclude that it is possible to implement claim validation as an information system with explicit trust in validation criteria. Based on the achievement of this dissertation, claim validation is expected to be implemented as an information system by explicitly defining the objects to be trusted according to the purpose and utilizing multiple evidence.

Keywords: 1. Claim Validation, 2. Trust, 3. Validity, 4. Digital Certificate, 5. Digital Signature

Keio University Graduate School of Media and Governance

Ryosuke Abe

学位審査委員会

主査

植原 啓介 (慶應義塾大学)

副査

楠本 博之 (慶應義塾大学)
服部 隆志 (慶應義塾大学)
中村 修 (慶應義塾大学)
村井 純 (慶應義塾大学)
鈴木 茂哉 (慶應義塾大学)

Academic Degree Evaluation Committee

Supervisor

Keisuke Uehara (Keio University)

Co-Supervisors

Kusumoto Hiroyuki (Keio University)
Takashi Hattori (Keio University)
Osamu Nakamura (Keio University)
Jun Murai (Keio University)
Shigeya Suzuki (Keio University)

目次

図目次

表目次

第 1 章

序論

1.1 背景

インターネットを通じて，人々は様々な情報をやり取りしながら生活している．情報システムを活用し，情報をやり取りすることで，人々はその地理的関係にとらわれずグローバルな空間中で協調し，様々な活動が可能となった．本研究では，複数のエンティティが情報をやり取りしながら協調して活動することを “協調プロセス (collaborative process)” と呼ぶ．例えば，遠隔地に対してインターネットを通じて教育コンテンツを提供することで，地理的関係にとらわれず様々な講義を受けられるサービスが一般に活用されている．講義は，講師からの一方的な遠隔講義のみならず，理解度を確認するための課題としてレポートの出題，および受講者によるレポートの提出を含む協調プロセスである．人々は，協調プロセスの結果として様々な実績を獲得し，当該実績を将来の活動の中で活用する．例えば，“特定の教育コンテンツを受講した” ことを実績として，就職活動の中で自身の学習歴として示すことが挙げられる．

こうした協調プロセスの中で，やり取りされる情報の検証可能性に関する課題が盛んに議論されている．例えば，大規模言語モデルの登場により大量に生成される偽情報への対策，システムに対する多様な攻撃への対応として逐次細かな単位で認証/認可する “ゼロトラスト” と呼ばれるアクセスコントロールモデルや，個人の実績の詐称やなりすましの対策としての証明書の活用が挙げられる [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]．これらの議論では，主に，デジタル署名などのデータセキュリティ技術が活用されている．例えば，アイデンティティ[*1]に関連する情報をデジタルデータで表現した “デジタル証明書” として発行，保持，提示するエコシステムが議論されている [13, 14, 15, 16, 17]．W3C では，2022 年にデジタル証明書のデータモデルである Verifiable Credentials Data Model v1.1

[*1] 本研究において，アイデンティティは “エンティティにまつわる属性情報の集合 (set of attributes related to an entity)” とする [12]

が，2025 年には v2.0 が標準化された [14, 15]．日本でも，COVID-19 ワクチン接種証明書が Verifiable Credentials ベースで発行され，デジタル庁の提供するスマートフォンアプリ上で保持，提示が可能であった [18]．学習歴にまつわる証明書の規格である Open Badges も，Spec Version 3.0 は Verifiable Credentials ベースとなっている [19, 20]．デジタル証明書は，典型的には暗号学的な手法を用いて，証明書自体が発行時点から改ざんされていないことを示す完全性と，証明書発行者が確認可能である．

デジタルな情報処理の利点の 1 つは，特定のデータで示される情報の集合を分解，あるいは他の情報と統合しながら，再利用できることにある．また，デジタル表現された情報は容易にデータとして複製可能であるため，提示先ごとに個別に用意することなく，都度複製されたデータを提示することで繰り返し提示可能である．デジタル証明書においても，物理的な紙の証明書と異なり，自身の保有する証明書を 1 度提示するだけでなく，様々な場面で繰り返し提示する形での活用が議論されている [21, 22, 23]．Verifiable Credentials 標準でも複数の証明書を提示するため，複数の証明書を内包した Verifiable Presentation として提示するデータモデルが規定されている．このことからも，単一の証明書の利用だけではなく，複数の証明書を組み合わせる利用が想定されていることが伺える．さらに，証明書自体の完全性を確認可能な状態を維持しつつ，一部の情報を秘匿しながら提示可能にする “選択的開示” の手法が議論されている [24, 25]．ここで，選択的開示は，単一の証明書中に示される複数の情報を分解しながら提示する手法と整理できる．

1.2 問題提起

本研究では証明書などで示され，特定のエンティティが主張する情報である “クレーム” の妥当性検証に着目する．前節で述べたように，証明書を様々な場面で活用するには，証明書で示されるクレームの真偽を含めた妥当性検証が重要である．しかし，証明書で示されるクレームの妥当性検証に関するモデルは，これまで明確に議論されていなかった．例えば，デジタル証明書の標準規格では，証明書の示すクレームの妥当性検証は検討の範囲外とされ，規定されていない [13, 14, 15]．そのため，典型的には特定の証明書の示すクレームの妥当性検証の基準は，当該証明書の活用が想定される特定のコンテキストに基づいて設計される．例えば，特定の証明書の提示によって，検証者は証明書の示すクレームを妥当であるとみなす場合，証明書発行者を “信頼” することで妥当性検証の基準を構成している，と整理できる．このとき，検証者は，“特定の証明書発行者が発行した証明書であること” を確認することで当該証明書の示すクレームを妥当とみなす．一方，“信頼” の概念は様々な学問分野で議論されているが，本研究の着目するクレームの妥当性検証において，情報システム観点からの整理はされていない [26]．すな

わち，このときなぜ “証明書発行者を信頼する” ことで，当該クレームを妥当等みなせるかを説明可能な “信頼” の概念の整理はされていない．検証者ごとのコンテキストを尊重し，クレームの妥当性検証を情報システムで実現するには，“信頼” が何を意味するのかを，情報システム観点から捉える必要がある．

信頼の概念をはじめとして，クレームの妥当性検証のモデルが存在しないため，多くの場合，クレームの検証基準にはコンテキスト特有の仮定や条件が前提として組み込まれる．一方，異なるコンテキストで当該証明書を活用する場合，元々のコンテキストにおける前提を暗黙的に援用してしまい，妥当性検証の目的に応じた適切な検証基準を構成できない恐れがある．それぞれのコンテキストに応じた適切な検証基準が構成されなければ，当該コンテキストでは許容できないクレーム誤認リスクを生じる可能性がある [27]．例えば，公開鍵証明書は広くインターネット上でエンティティと対応する公開鍵の紐付きを検証する方法として活用されている．典型的には，特定の証明書発行者が真正なクレームを証明書に記載することを仮定した上で，証明書の完全性検証に加え，当該発行者が発行した証明書であることを確認する．しかし，人為的なミスあるいはシステムへの攻撃などの要因により，発行者による紐付きの確認が十分されないままに証明書が発行された事例がある [28, 29, 30]．これらの事例では，発行者の発行基準とその運用を暗黙的に仮定し，検証者が証明書の完全性と発行者を確認するのみでは，検証者が適切に対象エンティティと公開鍵の紐付きを確認できず，誤認のリスクが生じる．

このように，妥当性検証において確認するべき項目や置くべき前提は，想定される誤認のリスクなど，クレームを妥当性検証するコンテキストに依存する．しかし，クレームの妥当性検証がデジタル証明書の標準規格の対象外とされていることからも，既存の取り組みにおいてコンテキストに応じたクレームの妥当性検証のモデルが十分に議論されているとは言いがたい．この問題は，学習歴を示すデジタル証明書の活用において，自国とは教育制度が異なる他国で発行された証明書を適切に解釈できないなど，証明書の相互運用性の欠如という形で顕在化している [31]．

1.3 手法

クレームの妥当性検証を様々なコンテキストの中で適切に実施するには，それぞれのコンテキストに併せた検証基準を構成することが不可欠である．そのためには，主張されるクレームの妥当性を，複数の根拠情報を活用しながら総合的に判断するモデルが必要である．情報システム視点での “信頼” の概念を整理し，クレームの妥当性検証のモデルが構成できれば，インターネットを通じた協調プロセスの中でやり取りされる情報の検証可能性を確保可能であると期待できる．

本研究では，主張者が特定のクレームを複数の根拠情報に基づいて主張し，検証者が当

該クレームの妥当性を検証するモデルを “情報システムにおけるクレームの妥当性検証” として提案する．モデルの提案にあたり，本研究では 4 項目の取り組みをおこなった．

1 つ目は，既存のデジタル証明書の標準規格で対象外とされている “クレームの妥当性検証” にまつわる，主張者と検証者間でのやり取りのモデル化である．まず，複数の標準文書を参照しながら関連概念を整理した上で，証明書などを用いたやり取りを情報システムに依存しない形で整理した．本研究においてはクレームの妥当性検証を，検証者がクレームを妥当とみなす検証基準を含む “検証ポリシ” に対し，主張者のクレームが適合するかどうかを判定するやり取りであるとした．

2 つ目は，1 つ目の取り組みによるモデルの中で，検証者による検証ポリシに応じた判定を担う情報システムを構成する手法の提案である．情報システムとしてクレームの検証を実現するために，検証ポリシに沿って，検証対象のクレームと根拠情報を入力とし，検証結果を出力する関数モデルを定義した．また，検証ポリシで示される検証基準に，計算機で決定困難な命題を含む可能性があることから，仮定を導入することをクレームの妥当性検証における “信頼” であると整理した．そこで，信頼に基づいてコンテキストに沿った検証基準を構成する抽象モデルである “Shinken モデル” を提案した．

続く 2 つの取り組みでは，Shinken モデルを適用したクレームの妥当性検証の実証として，2 件のケーススタディを実施した．1 件目は，商取引において，取引相手の不誠実な振る舞いによる経済的損失を回避するため，取引相手の過去の取引結果をクレームとして検証することで，リスク推定の礎とする取り組みである．2 件目は，デジタル証明書を用いたクレームの妥当性検証において，コンテキストに合わせた検証基準を構成するために，検証基準を更改する取り組みである．

以上の取り組みによって，情報システムとして複数の根拠情報を元にクレームの妥当性検証を実現するモデルを提案し，その適用可能性を評価する．

1.4 本研究の貢献

先述の取り組みを通じて，本研究は以下の点において貢献を果たす．

- **クレーム妥当性検証のモデルの提示**: 従来のデジタル証明書の標準規格では対象外とされてきたクレームの妥当性検証を，検証者の意図を反映した “検証ポリシ” で示される検証基準への適合を判断する営みとしてモデル化
- **信頼に基づき妥当性検証の基準を構成する手法の提案**: “信頼” の概念を妥当性検証の基準の構成に導入し，透明性があり更改可能な検証基準を構成する “Shinken モデル” の提案
- **ケーススタディによる実証**: 2 つのケーススタディを通じて，異なる文脈において Shinken モデルに基づきクレームの妥当性検証を情報システムとして実装可能で

あることを実証

以上を総括し，クレームの妥当性検証を情報システムとして実現可能なモデルを確立したことが，本研究全体の貢献である．

1.5 本論文の構成

本論文は，以下のように構成されている．2 章では，1 つ目の取り組みとして，クレームの妥当性検証のモデルを定義することで，人々が情報をやり取りする中で如何にその妥当性を主張し，検証しているかを議論する．3 章では，本研究の議論に登場する要素技術を概説する．4 章では，2 つ目の取り組みとして，計算機を用いた情報システムにおいて，クレームの妥当性検証を実装するための関数モデルと，その課題を定義する．5 章では，4 章で述べた課題に対処し，情報システムでクレームの妥当性検証を実現するための “Shinken モデル” を提案する．続く 6 章と 7 章では，Shinken モデルを適用したケーススタディによって，Shinken モデルの適用可能性を議論する．1 つ目のケーススタディとして，6 章では，商取引において，将来の取引相手が取引開始前に取引中の紛争リスクを推定可能にするため，過去の取引結果をクレームとして主張，検証するケースを取り上げる．2 つ目のケーススタディとして，7 章では，デジタル証明書を用いたクレームの妥当性検証において，仮定が破られた場合を想定し，検証基準を更改するケースを取り上げる．8 章では，6 章と 7 章で議論するケーススタディをもとに，Shinken モデルの適用可能性を評価し，その有用性を議論する．最後に，9 章で，本研究の議論を総括する．

第 2 章

クレームの妥当性検証のモデリング

本章では，本研究におけるクレームの妥当性検証のやり取りのモデルを概説する．まず，日本語の “検証” にあたる英語の概念として “Validation” と “Verification” について整理する．次に，モデル中で登場するエンティティおよびその概念について整理し，主張者と検証者の 2 者間でのやり取りのモデルとして定義する．なお，本章で議論するモデルは情報システムに依存しない，エンティティ間のやり取りのモデルとして整理する．

2.1 Verification と Validation

情報の検証可能性を議論するにあたり，日本語の “検証” に相当する概念を示す英単語として “Verification” と “Validation” がある．本節では，2 つの概念の差異にについて整理する．

Oxford English Dictionary では “Verify” は次のように定義されている [32].

> *To testify to, to assert, to affirm or confirm, as true or certain.*

したがって，“Verify” とは，対象が真であるかどうかを確認する行為である．一方，“Validate” は次のように定義されている．

> *To make valid or of good authority; to confirm or corroborate; to substantiate or support.*

ここでは，“Validate” には，対象が “妥当（valid）” であるかどうかの判断が含まれると整理する．したがって，判断するエンティティ，すなわち検証者が持つ判断の目的や意図を含むコンテキストが重要な役割を果たす．そこで，本研究では，検証者による対象を確認するコンテキストにその結果が依存するかどうかが “Verification” と “Validation” の違いであると整理する．すなわち，コンテキストに関わらず，対象の示される形式などを確認する行為が “Verification” であり，コンテキストを尊重し，対象の妥当性を確認する

行為が “Validation” である．次に，ISO 9000 と Verifiable Credentials の 2 つの標準における定義を精査し，同様の整理が可能かどうかを確認する．

2.1.1 ISO 9000 における定義

ISO 9000 は，品質マネジメントシステムの基礎と用語を定義する標準である [33]．複数の NIST の文書における “Verification” と “Validation” の定義は，ISO 9000 を参照して定義されている [34, 35, 36]．ISO 9000 における “Verification” は次のように定義されている．

> *Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled*

また，“Validation” は次のように定義されている．

> *Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled*

双方において “objective evidence” に基づき “requirements” が満たされるかどうかを確認することが，それぞれの行為であると定義されている．一方，それぞれにおいて “requirements” の特性が異なる．“Verification” は，単に “specified requirements” が満たされかどうかを確認する行為である．また，“Validation” は，“specified requirements for a specific intended use or application” が満たされたかどうかを確認する行為である．“Validation” における “specific intended use or application” は，その行為主体の目的や意図を反映していると考えられ，確認行為のコンテキストに依存すると整理できる．したがって，Oxford English Dictionary の定義に沿って整理したように，“Validation” においてはそのコンテキストが重要な役割を果たすと考えられる．

2.1.2 Verifiable Credentials Data Model v2.0 における定義

次に，デジタル証明書の標準である Verifiable Credentials Data Model v2.0 における定義を議論する [15]．“Verification” は次のように定義されている．

> *The evaluation of whether a verifiable credential or verifiable presentation is an authentic and current statement of the issuer or presenter, respectively.* (本論文著者により中略) *Verification of a credential does not imply evaluation of the truth of claims encoded in the credential.*

また，“Validation” は次のように定義されている．

> *The assurance that a claim from a specific issuer satisfies the business requirements of a verifier for a particular use.* (本論文著者により中略) *However, the means for such validation vary widely and are outside the scope of this specification. It is expected that verifiers will trust certain issuers for certain claims and apply their own rules to determine which claims in which credentials are suitable for use by their systems.*

すなわち，“Verification” は，デジタル証明書の発行者あるいは提示者の確認であるとされ，証明書で示されるクレームの真偽は当該標準の範囲外であると定義されている．証明書の発行者および提示者は，検証者が誰であるかには無関係である．そのため，“Verification” の結果は検証者のコンテキストには依存しない．“Validation” は，検証者の視点で，証明書に示されるクレームが要求を満たすかどうかの確認であるとされている．異なる検証者はそのコンテキストに応じてそれぞれ異なる要求を持つため，それぞれの “Validation” の結果は異なると考えられる．したがって，Verifiable Credentials における “Validation” 定義においても，Oxford English Dictionary および ISO 9000 と同じく，コンテキストが重要な役割を果たす．

2.1.3 小括

本節では，日本語における “検証” に対応する英語の概念として “Validation” と “Verification” を取り上げた．Oxford English Dictionary の定義を参照し，本研究では “検証者による対象の確認にかかる意図や目的などのコンテキストに結果が依存するかどうかが，Validation と Verification の違いである” と整理した．ISO 9000，Verifiable Credentials の標準における定義を参照し，2 つの概念の差異を同様に整理可能か確認した．その結果，それぞれの定義において，本研究における整理と同様に “Validation” においてはそのコンテキストが重要な役割を果たすことが明らかとなった．本研究では，特に Verifiable Credentials の標準で範囲外とされている “Validation” に着目し，コンテキストに応じた情報の妥当性検証を議論する．

2.2 エンティティと概念

本節では，本研究で議論するモデルに登場するエンティティと概念を概説する．まず，以下の用語を定義する．

- **検証 (Validate)**: ある情報が妥当であることの確認
- **クレーム (Claim)**: あるエンティティが主張 (assert) する情報
- **根拠情報 (Evidence)**: あるクレームが妥当であることを示唆 (imply) する情報

したがって，本研究において，“クレーム検証” とは，クレームの妥当性検証（Claim Validation）を指す． エンティティとして，以下の 2 者を定義する．

- **主張者 (Claimant)**: あるクレームを主張するエンティティ
- **検証者 (Validator)**: 主張者によって主張されたクレームを検証するエンティティ

また，検証者がクレームを検証するにあたり，“妥当であること” を定義し，その基準を示す必要がある．一方，主張者も同様にクレームの主張の方法を定義する必要がある．したがって，検証者と主張者は，それぞれ以下のポリシ[*1]を定義する．

- **検証ポリシ (Validation Policy)**: 検証者が特定のクレームをどのように検証するかを定義するポリシ
- **主張ポリシ (Assertion Policy)**: 主張者が特定のクレームをどのように主張するかを定義するポリシ

ここまでの概念整理に基づき，本研究において検証は，検証ポリシに示される検証基準に従って検証者が妥当であると判定する行為である．そのため，クレームで示される事象が検証者以外にとって認知された事実であるとは限らない．

アプリケーションの中でクレームを検証する時，検証者は検証した上で，当該クレームを何らかの形で活用する．例えば，主張者のアイデンティティに関わる情報を証明書の形式でクレームとして受け取り，検証した上で，主張者に対してアプリケーション中の権限を与えることが考えられる [37, 38, 39]．検証されたクレームをアプリケーションの中でどのように活用するかというコンテキストに応じて，検証者にとってどの程度の確度でクレームを検証するかを決定し，検証ポリシに検証基準を定める．

2.3 やり取りのモデル

本節では，前節で定義した概念を用いて，クレームの主張と検証のやり取りのモデルを定義する．まず，主張者と検証者の 2 者間のやり取りを “基本モデル” として整理する．次に，検証者が直接検証できないクレームを扱うために，証明書を活用することで間接的にクレームを検証する “証明書モデル” を導入する．

[*1] 本研究では，ポリシを「意思決定の基礎として使用される，規則などの一連の考え方やルール」と定義する．

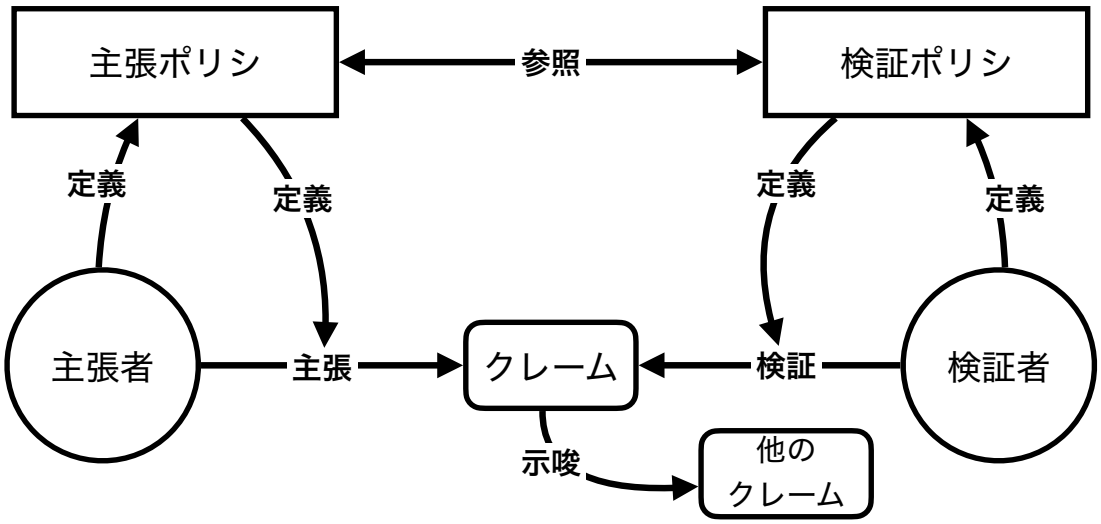

図 2.1　基本モデル: 主張ポリシと検証ポリシの相互作用によって，特定のクレームを主張あるいは検証する方法が決定される．また特定のクレームは他のクレームを示唆するケースも考えられ，これにより多様なクレームの主張あるいは検証を実現する．

2.3.1　基本モデル

基本モデルとして，主張者と検証者の 2 者間におけるクレームのやり取りを議論する．図 2.1 にモデルの概要を示す．主張者は，検証者の検証ポリシを参照することで，検証者がどのような検証基準でクレームを検証するのかを確認し，どのようにクレームを提示するかを決定する．例えば，主張者の本人確認のために，検証者が特定の公的身分証明書の提示を検証ポリシ中で指定するケースが考えられる．この時，主張者は，検証者の検証ポリシを参照することで，検証者が受け入れる証明書を確認した上で，該当する証明書を提示する．一方，主張者が特定の方法で主張することが明らかであるために，当該主張方法を検証者が受け入れる，というケースも考えられる．したがって，検証ポリシと主張ポリシは相互に作用し合うことで，どのようにやり取りをするかの具体的なプロトコルが規定される，と考えられる．

また，あるクレームは，他のクレームを示唆 (imply) するケースも考えられる．この特性により，複数のクレームを根拠情報として組み合わせることで，多様なクレームを主張あるいは検証する，という営みが成立する．たとえば，“特定の大学の学部/学科を卒業した” というクレームは，“当該学部/学科で扱う分野への専門性を持つ” というクレームを示唆する．一方で，真に専門性を持つかどうかは，当該学部/学科で具体的にどのような教育を受けたか等によって影響を受けるため，後者のクレームは容易に断定できない．したがって，検証者は，複数のクレームを根拠情報として組み合わせ，目的のクレームを妥当であるとみなす検証基準を，検証ポリシの中に定める．

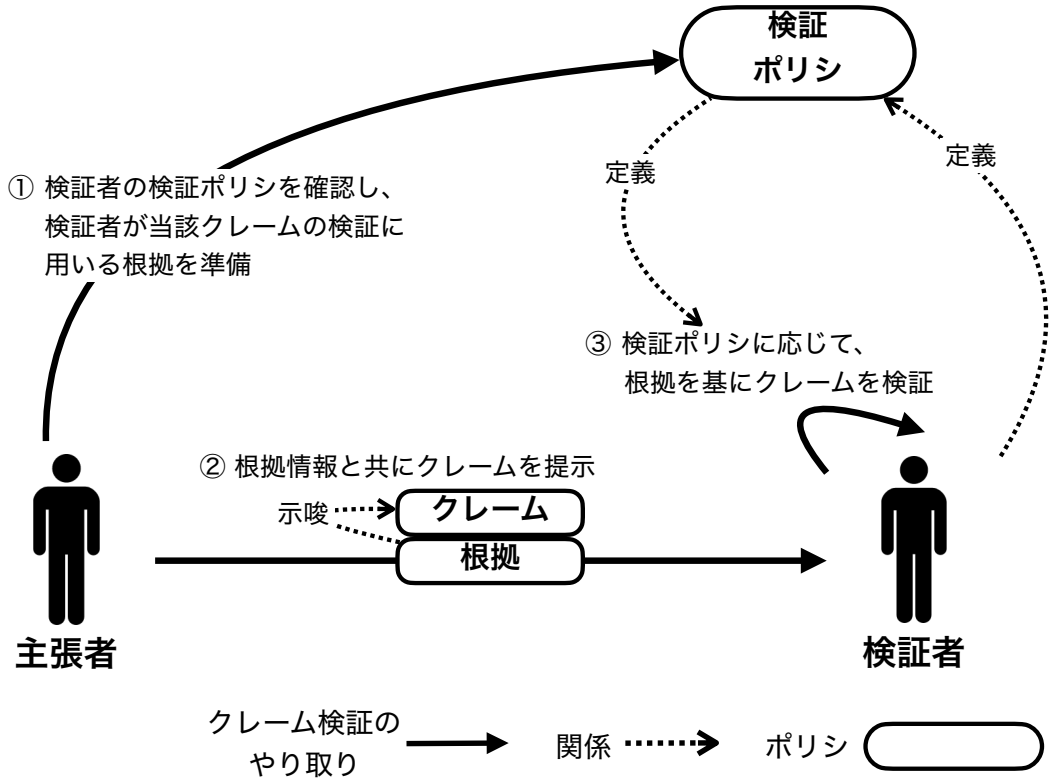

図 2.2　基本モデルを基にしたやり取りの概要: 主張者は，検証者の検証ポリシを参照し，検証者がどのようにクレームを検証するかを確認する．その上で，主張者は根拠情報とともにクレームを提示し，検証者は検証ポリシに沿ってクレームを検証する．

図 2.2 に基本モデルを基にしたやり取りの概要を示す．まず，主張者は特定のクレームを検証者に検証させるため，検証者の検証ポリシを確認し，検証者が当該クレームの検証のために用いる根拠情報を確認する．次に，主張者は検証ポリシで定められた根拠情報を準備し，クレームとともに検証者へ提示する．検証者は提示されたクレームと根拠情報を基に，自身の検証ポリシで示した検証基準に応じて，クレームを検証する．

2.3.2　基本モデルの限界

本節では，基本モデルの限界について議論する．基本モデルでは，主張者が根拠情報を検証者へ提示する．しかし，根拠情報が特定のシステム内にサイロ化されており，主張者自身で提示できないケースが考えられる．また，プライバシの観点より，根拠情報を検証者に開示すること自体が望ましくないケースも考えられる．さらに，開示可能であったとしても，検証者の専門性の欠如により，根拠情報が対象のクレームを示唆することを確認できないケースが考えられる．

したがって，基本モデルを適用するためには，検証者視点で以下の要件を満たす必要が

ある．

- 根拠情報が検証者にとってアクセス可能であること
- 根拠情報が特定のクレームを示唆することを検証者が確認可能であること

両者が満たせない場合，検証者はクレームを検証できず，基本モデルは直接適用できない．

2.3.3 証明書モデル

前節で述べた限界へ対処として，複数の基本モデルを組み合わせることで，間接的にクレームを検証できる．本節では，複数の基本モデルの組み合わせの一例として，証明書を用いたモデルを概説する．証明書モデルでは，以下の 2 種類のクレームを組み合わせる．

- **対象クレーム (Subject Claim)**: 検証者が検証する対象のクレーム．検証者は直接本クレームを検証できない．
- **認定クレーム (Certification Claim)**: “対象クレームを検証した” ことを示すクレーム．本クレームは，検証済みである対象クレームが妥当であることを示唆する．

また，新たに以下のエンティティを定義する．

- **認定者 (Certifier)**: 対象クレームを検証し，認定クレームを主張するエンティティ

したがって，証明書は以下のように定義できる．

- **証明書 (Certificate)**: “特定の認定者が対象クレームを検証した” という認定クレームを示すもの

証明書モデルでは，3 者の間でそれぞれが各クレームを主張および検証することで，検証者は目的とする対象クレームを検証できる．対象クレーム s および認定クレーム c に対して，主張者 $\mathbb{A}$，検証者 $\mathbb{V}$，認定者 $\mathbb{C}$ が定義する主張ポリシ (AP) および検証ポリシ (VP) を以下のように定義する[*2]．

- $\mathsf{AP}_s^{\mathbb{A}\to\mathbb{V}}$: 主張者 $\mathbb{A}$ が，対象クレーム s を検証者 $\mathbb{V}$ に主張するための主張ポリシ
- $\mathsf{AP}_s^{\mathbb{A}\to\mathbb{C}}$: 主張者 $\mathbb{A}$ が，対象クレーム s を認定者 $\mathbb{C}$ に主張するための主張ポリシ

[*2] 本研究における “主張者” は英語で “Claimant” であるが，認定者である “Certifier” との混同を避けるため，本節では，主張者はクレームを主張する (assert) エンティティであることから “Asserter” の頭文字より $\mathbb{A}$ とした．

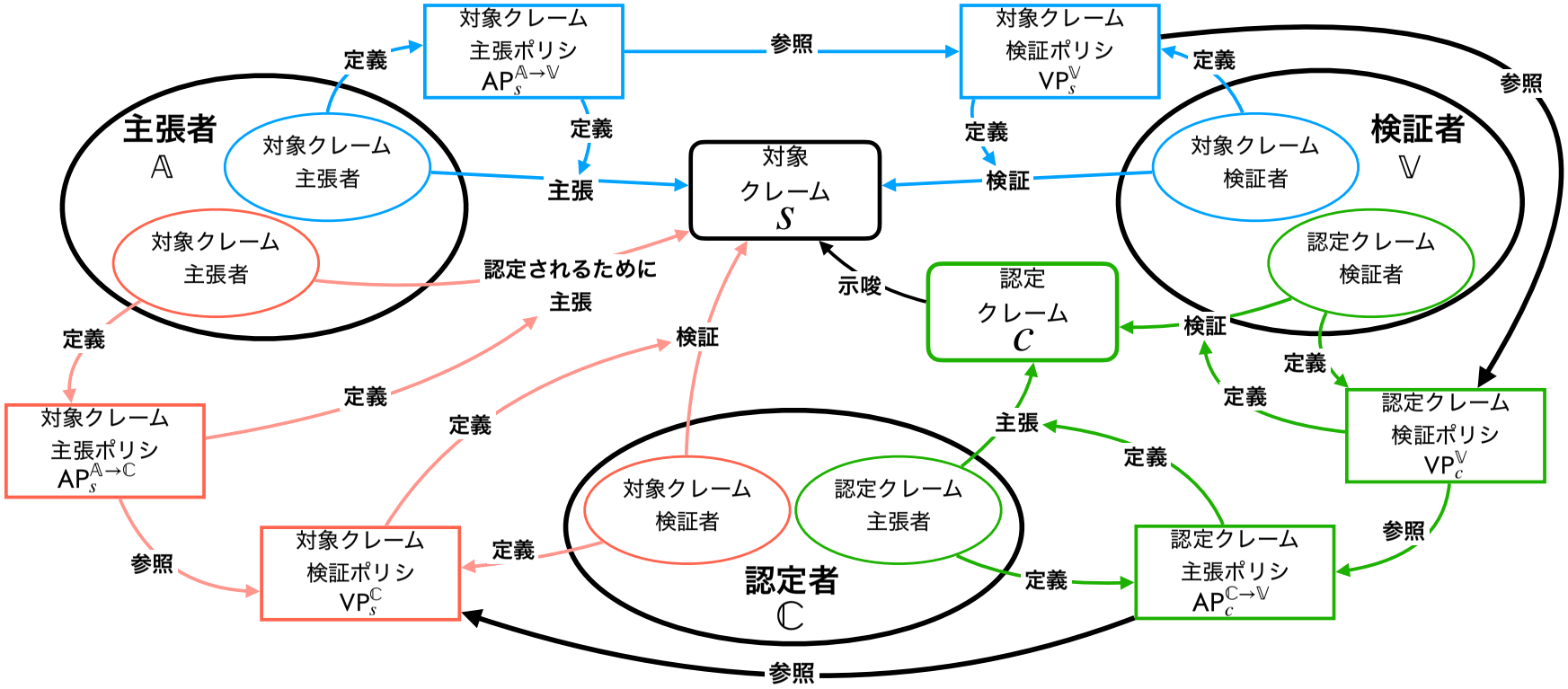

図 2.3　証明書モデル: 検証者は検証対象のクレームを直接検証できない場合，認定者による“対象クレームを認定者が検証した”ことを示す認定クレームを検証することで，間接的に対象クレームを検証できる．この時，検証者は認定クレームが対象クレームを示唆しているとみなす．

- $\mathsf{AP}_c^{\mathbb{C}\to\mathbb{V}}$: 認定者 $\mathbb{C}$ が，認定クレーム c を検証者 $\mathbb{V}$ に主張するための主張ポリシ
- $\mathsf{VP}_s^{\mathbb{V}}$: 検証者 $\mathbb{V}$ が，対象クレーム s を検証するための検証ポリシ
- $\mathsf{VP}_s^{\mathbb{C}}$: 認定者 $\mathbb{C}$ が，対象クレーム s を検証するための検証ポリシ
- $\mathsf{VP}_c^{\mathbb{V}}$: 検証者 $\mathbb{V}$ が，認定クレーム c を検証するための検証ポリシ

まず，検証者は，対象クレームを検証するために，特定の認定者から発行される証明書を用いることを対象クレーム検証ポリシ $\mathsf{VP}_s^{\mathbb{V}}$ に定める．主張者は当該検証ポリシ $\mathsf{VP}_s^{\mathbb{V}}$ を参照し，該当する証明書を入手するために，認定者に対して対象クレームを主張する．認定者は，対象クレームを直接検証できることから，主張者と認定者の間は基本モデルのやり取りが成立する．主張者は，認定者の対象クレーム検証ポリシ $\mathsf{VP}_s^{\mathbb{C}}$ を参照し，認定者向けの対象クレーム主張ポリシ $\mathsf{AP}_s^{\mathbb{A}\to\mathbb{C}}$ に基づいてクレームを主張する．

つぎに，認定者は，“自身が対象クレームを検証した” ことを認定クレームとして主張する．この時，認定者は，自身の対象クレーム検証ポリシ $\mathsf{VP}_s^{\mathbb{C}}$ を参照し，“検証ポリシに沿って対象クレームが検証できた” ことをクレームとして主張するための，認定クレーム主張ポリシ $\mathsf{AP}_c^{\mathbb{C}\to\mathbb{V}}$ を定義する．また，検証者は，認定者の認定クレーム主張ポリシ $\mathsf{AP}_c^{\mathbb{C}\to\mathbb{V}}$ を参照しながら，認定クレーム検証ポリシ $\mathsf{VP}_c^{\mathbb{V}}$ を定義する．さらに，検証者自身の対象クレーム検証ポリシ $\mathsf{VP}_s^{\mathbb{V}}$ の中で認定クレーム検証ポリシ $\mathsf{VP}_c^{\mathbb{V}}$ を参照する．これによって，認定クレームを検証することで，対象クレームを検証したとみなすことを示す．

証明書モデルは，検証者の対象クレーム検証ポリシ $\mathsf{VP}_s^{\mathbb{V}}$ からの各ポリシの参照の連鎖によって成立する．図 2.4 に証明書モデルにおけるポリシの関係性を示す．証明書モデルにおいては，各ポリシの参照関係が示すポリシで示される情報の関係性を，演繹関係とみなすことで，対象クレームの検証が成立する．また，検証者の検証ポリシは連鎖する各ポリシを内包しているとみなせる．したがって，クレーム検証は，検証者の検証ポリシで規定される検証基準をどのように構成し，当該基準に適合するかどうかという判定問題に帰着する．

図 2.5 に証明書モデルを基にしたやり取りの概要を示す．検証者は，自身の検証ポリシで特定の認定者が発行する証明書を根拠情報としてクレームを検証することを定義する．主張者は，当該検証者の検証ポリシを確認し，根拠情報となる証明書の発行を認定者に要求する．証明書の発行にあたり，主張者と認定者の間では基本モデルに準じて当該クレームの主張と検証を実施する．認定者がクレームを検証できれば，認定者は “当該クレームが検証できた” ことを示す証明書を主張者へ発行する．次に，主張者は当該証明書を根拠情報として検証者に提示する．検証者は，自身の検証ポリシに沿って当該証明書が特定の認定者から発行されていることを確認し，クレームを検証する．

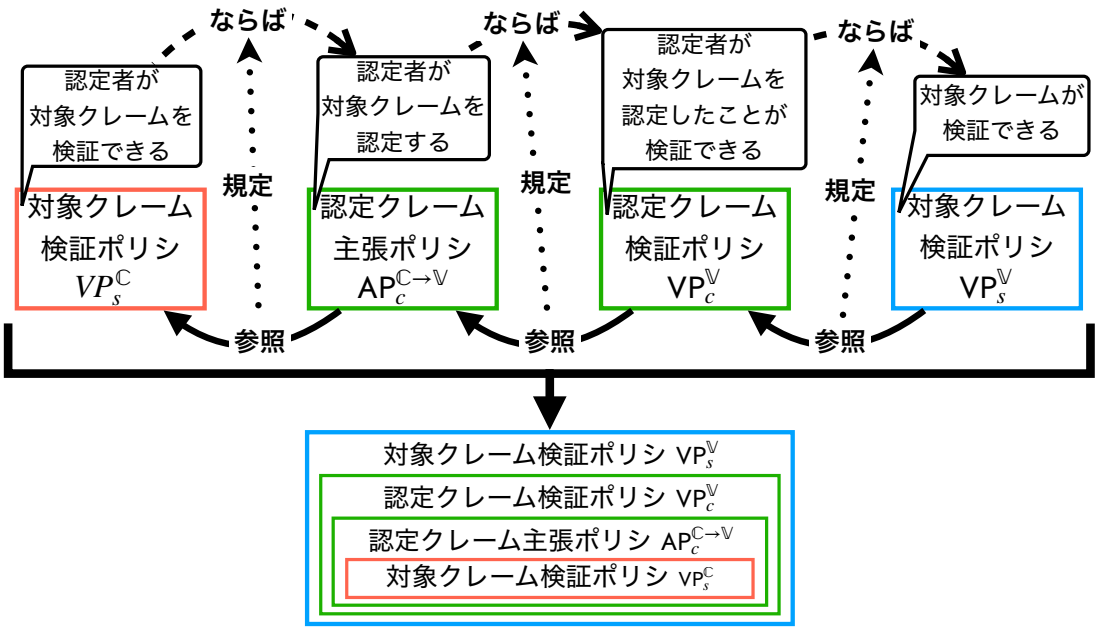

図 2.4　証明書モデルにおけるポリシの関係性: 各ポリシによって示される情報の関係性を，ポリシの参照関係から演繹関係とすることで，対象クレーム検証が実現される．したがって，検証者による検証ポリシは，参照する各ポリシを内包するとみなせる．

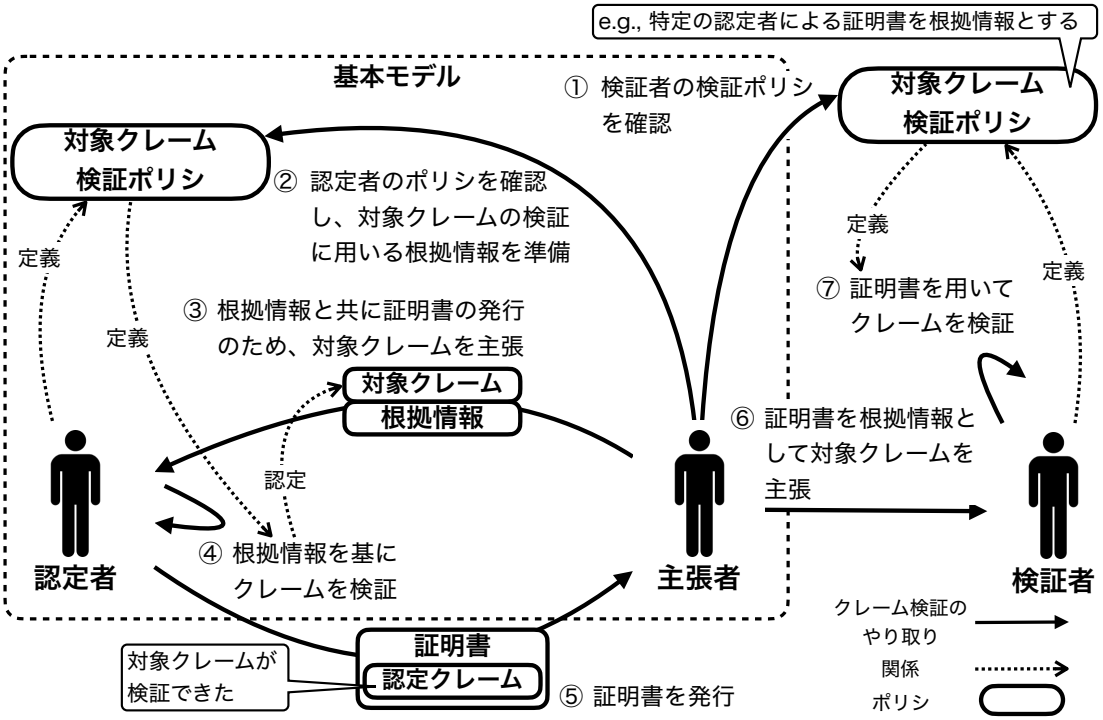

図 2.5　証明書モデルを用いたやり取りの概要: 検証者は，認定者による証明書を根拠情報とすることを自身の対象クレーム検証ポリシに定める．そこで，主張者は，基本モデルのやりとりに基づき認定者へ証明書の発行を要求する．認定者はクレームの検証したのち，認定クレームを示す証明書を発行する．主張者は当該証明書を検証者へ提示し，検証者は証明書を用いて間接的に対象クレームを検証する．

2.4 クレーム検証に関する既存の議論

本節では，本章で議論したモデルに基づき関連する既存の議論を概説する．Nabil らは，理論的なモデルとして一般的な事象に対するクレーム検証を議論した [40]．Nabil らのモデルでは，複数の根拠情報から，根拠情報がクレームを示唆する確率を計算する確率関数としてクレーム検証をモデル化した．例えば，“全ての白鳥は白い” というクレームに対して，“白い白鳥が観測された” という事実を根拠情報として，一般的な事象が事実である確率を計算する．本章で議論したモデルに基づくと，Nabil らのモデルは，検証ポリシにおいて確率関数でどのように確率を算出するかを定義していると整理できる．

アプリケーションの観点からは，様々な観点でクレーム検証が議論されている．例えば，学術論文の査読は，論文の著者が主張者として論文にクレームを記載し，論文誌の査読ポリシとして規定される検証ポリシに従って，査読者が検証者として当該クレームを検証する仕組みであると整理できる．論文で示されるクレームの根拠情報として，論文自体に加え，実験データを投稿時に併せて提示することで，クレーム検証をより容易かつ確度を高く実施する議論がある [41]．また，査読のプロセスを機械学習などの手法を適用しながら自動化する議論がある [42, 43, 44]．これらの議論は，査読ポリシに定められる論文誌掲載の基準に沿って，当該論文の示すクレームが妥当であるかどうか判断するシステムを構成する試みであると整理できる．

Web Credibility と呼ばれる，Web 上の記事の信憑性を確認する議論がある [45]．図 2.6 に Web Credibility におけるクレーム検証の概念図を示す．これらの議論は，記事の著者や掲載箇所，デザインなどの対象記事に含まれる，あるいは参照される情報を根拠情報として，当該記事が示すクレームを検証する議論であると整理できる．つまり，本章で議論したモデルに基づくと，記事の著者が主張者であり，記事の閲覧者が検証者として，閲覧者の基準に基づいてその記事で示されるクレームの妥当性を検証するやり取りと整理できる．

2.5 本章のまとめ

本章では，クレーム検証をモデリングし，主張者と検証者の間のやり取りのモデルとして整理した．まず，日本語の “検証” に相当する英語の概念として，“Verification” と “Validation” の差異を整理し，“Validation” は検証者のコンテキストを尊重した対象の妥当性の確認であると整理した．そこで，クレームの妥当性検証を “クレーム検証（Claim Validation）” と定義し，そのモデルを整理した．クレーム検証では，検証者と主張者それぞれが検証ポリシ，主張ポリシを規定し，それらに沿った形でクレームを主張および検証する．各ポリシは，相互に関係し合うことによって，どのような根拠情報を提示し，クレームを検証できたとみなすかが決定される．証明書モデルでは，検証者が直接

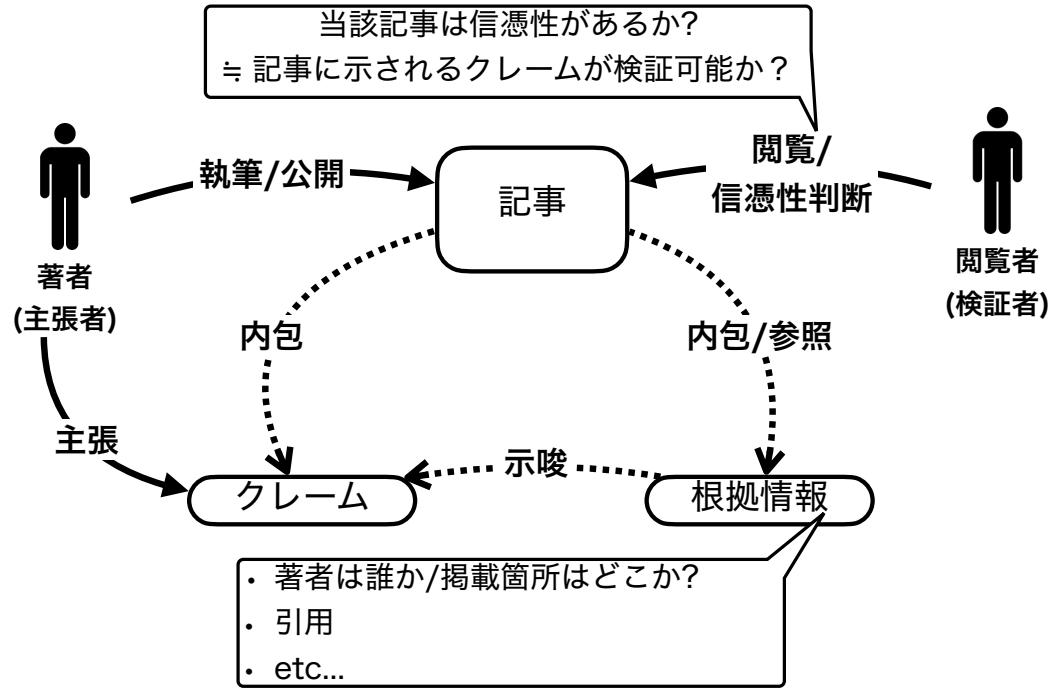

図 2.6　Web Credibility におけるクレーム検証の概念図: 記事の著者が主張者として，クレームを含む記事を公開する．著者，公開されたドメイン，あるいは記事のデザインなどの情報を根拠として，閲覧者は検証者として当該クレームを検証する．

検証できないクレームに対して，認定者が対象クレームを直接検証した，という別のクレームが対象クレームを示唆するために成立するものであると整理した．また，証明書モデルでは検証者，認定者それぞれのポリシの参照関係を，ポリシによって示される情報の演繹関係と整理できる．したがって，検証者自身の検証ポリシがそれぞれのポリシを内包している，とみなせることを議論した．最後に既存のクレーム検証に関連する議論を取り上げ，本章で議論したモデルに基づいて整理した．

次章では，本研究の議論に登場する要素技術を概説する．

第 3 章

要素技術

本章では，本研究の議論に関連する要素技術を概説する．まず，データの完全性を保証するためのデジタル署名および署名付きデータのモデルを概説する．次に，デジタル証明書の標準規格である Verifiable Credentials について述べる．また，本研究の議論の一部で活用するブロックチェーン技術についても述べる．

3.1 デジタル署名付きデータ

本節では，デジタル署名付きデータ (Signed Data) の概念を概説する．まず，デジタル署名の概念を概説し，次に署名付きデータを定義する．また，署名付きデータの例として，JSON Web Signature および JSON Web Token を概説する．

3.1.1 デジタル署名の概念

デジタル署名は，対象データが当該署名作成時点より完全性を保っていることを保証するデータである．典型的には，非対称暗号を用いて実装される．本研究では，非対称暗号における 2 つの鍵の役割を以下のように定義する．

- **署名鍵 (Signing key)**: 署名作成に用いる鍵
- **検証鍵 (Verification key)**: 署名検証に用いる鍵．検証鍵は，署名作成に用いられる署名鍵に対応する．

また，署名を作成および検証するために，以下の 2 つの操作を定義する[*1]．

[*1] 本研究において，“検証” は “Validation” を指し，デジタル署名の “検証” はその形式的な確認であるため “Verification” である．一方，“署名検証” および “検証鍵” は一般的に用いられている用語であることから，本研究においてもデジタル署名に対する操作は Verification であるものの例外的に “検証” の語を用いる．

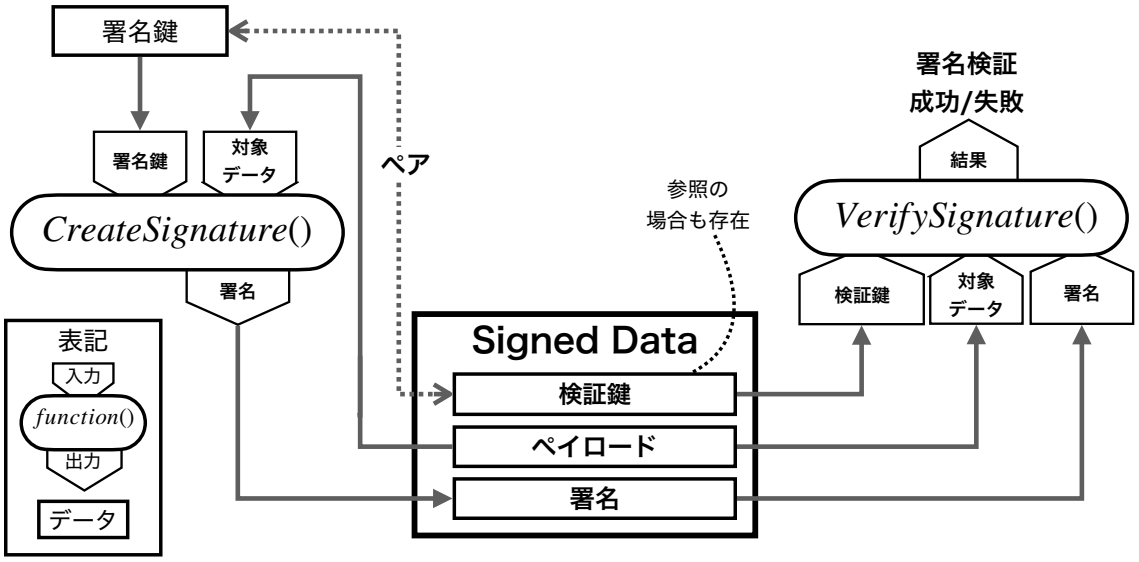

図 3.1　署名付きデータの概念データモデル: 署名付きデータは，1) 検証鍵，2) ペイロード，3) 署名から構成される．

- **CreateSignature(対象データ, 署名鍵)**: 署名鍵と対象データを入力とし，署名を出力する関数
- **VerifySignature(対象データ, 署名, 検証鍵)**: 署名を作成する際に使用された署名鍵と入力された検証鍵が一致しているか，および署名作成時点から対象データが完全性を保っているかを出力する関数

本研究では，RSA や ECDSA などの署名アルゴリズムが署名鍵，検証鍵，署名，および対象データの関係性を保証していると仮定し，具体的な署名アルゴリズムと独立な形で議論する [46, 47]．検証鍵があるエンティティのアイデンティティに結びつく時，署名は，対象データに当該エンティティが署名したことを示せる．

3.1.2　署名付きデータのデータモデル

デジタル署名付きデータは，署名者が対象データの完全性を主張するための概念データモデルである．図 3.1 に署名付きデータの概念データモデルを示す．署名付きデータは以下の要素から構成される．

- **検証鍵**: 署名に対する検証鍵，あるいは検証鍵の参照
- **ペイロード**: 署名者が完全性を主張する対象
- **署名**: ペイロードに対する署名

検証者が署名付きデータの完全性を検証する際は，検証鍵を取り出し，署名を検証す

る．また，検証鍵が署名者のアイデンティティと結びつくとき，検証者は署名者が当該ペイロードに対して署名を作成したことが確認できる．

3.1.3 デジタル署名付きデータの例: JSON Web Signature と JSON Web Token

本節では，デジタル署名付きデータの例として JSON Web Signature (JWS) を概説する [48]．また，特に JSON で表現されたデータに対して署名を付与し，アクセストークンなどとして活用する JSON Web Token (JWT) を概説する [49]．

JWS は，以下の要素から構成される．

- **JOSE Header**: ペイロードに対する暗号操作に関するパラメータを表現する JSON オブジェクト．鍵や，署名アルゴリズムなどが指定される．
- **JWS Payload**: 署名対象のデータ
- **JWS Signature**: ペイロードの完全性を保証するためのデジタル署名．シリアライゼーションの形式によっては，JOSE Header も署名対象に含まれる．

JWS の標準では，各要素を BASE64URL エンコードした上で "." で連結し URL-safe な文字列で表現する JWS Compact Serialization と，全体を JSON オブジェクトとして表現する JWS JSON Serialization の 2 つのシリアライゼーション形式が定められている． JSOE Header には検証鍵自体を含むか，その参照を示す値を指定できる． JWS Signature 検証時は指定された検証鍵を取り出し，署名を検証する．

JWT は，JSON で示されたデータを URL-safe な形式やり取りするためにシリアライズする方法を規定した規格である．JSON オブジェクトで示されたクレームに対して，JWS のペイロードとして BASE64URL エンコードし，デジタル署名を付与する．図 3.2 に JWT の生成例を示す．JWT の規格では，ペイロードとする JSON オブジェクトの中に含めるデータのフィールド名として，幾つかの利用用途が限定されたフィールド名を "Registered Claim Names" として規定している．例えば，当該 JWT の発行者を示す "iss" フィールドや，当該 JWT を受取ると期待されるエンティティを示す "aud" フィールドなどが規定されている．Registered Claim Names で指定されたフィールド名以外は，自由にフィールド名を記述可能である．

3.2 デジタル証明書関連技術

本章では，デジタル証明書の標準の一例として Verifiable Credentials を取り上げる．また，関連して，証明書の示す情報の分解，統合の手法として，証明書に示される情報を部分的に開示する手法である "選択的開示 (Selective Disclosure)" についても概説する．

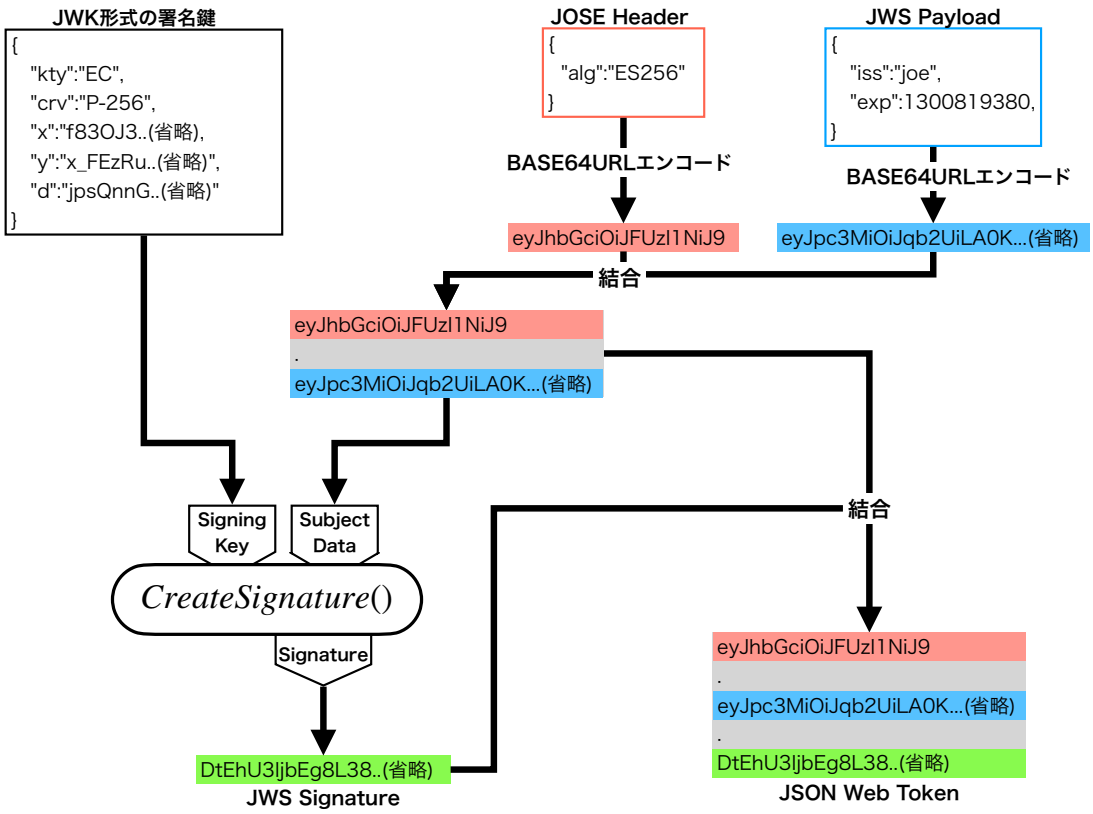

図 3.2　JSON Web Token の生成例: 本図では，JWS Compact Serialization で表現された JWT の生成例を示す.

3.2.1　Verifiable Credentials

Verifiable Credentials は，デジタル証明書を表現するためのデータモデルを規定する標準規格である [14, 15]．Verifiable Credentials のデータモデルでは，ある証明書発行者 (issuer) が証明書所有者 (holder) に対して証明書を発行し，証明書検証者 (verifier) に対して当該証明書を提示するモデルを規定している．Verifiable Credentials は以下の要素から構成される.

- **Credential Metadata**: 証明書発行者や，有効期限などの証明書自体に関する情報
- **Claim(s)**: 当該証明書が示すクレーム
- **Proof**: 当該証明書の完全性を示すための情報

典型的には，Verifiable Credentials は署名付きデータで表現され，証明書検証者は Proof に含まれるデジタル署名を検証することで，特定の証明書発行者が当該 Verifiable Credentials を発行したことを検証できる．例えば，証明書検証者は，Credential Meta-

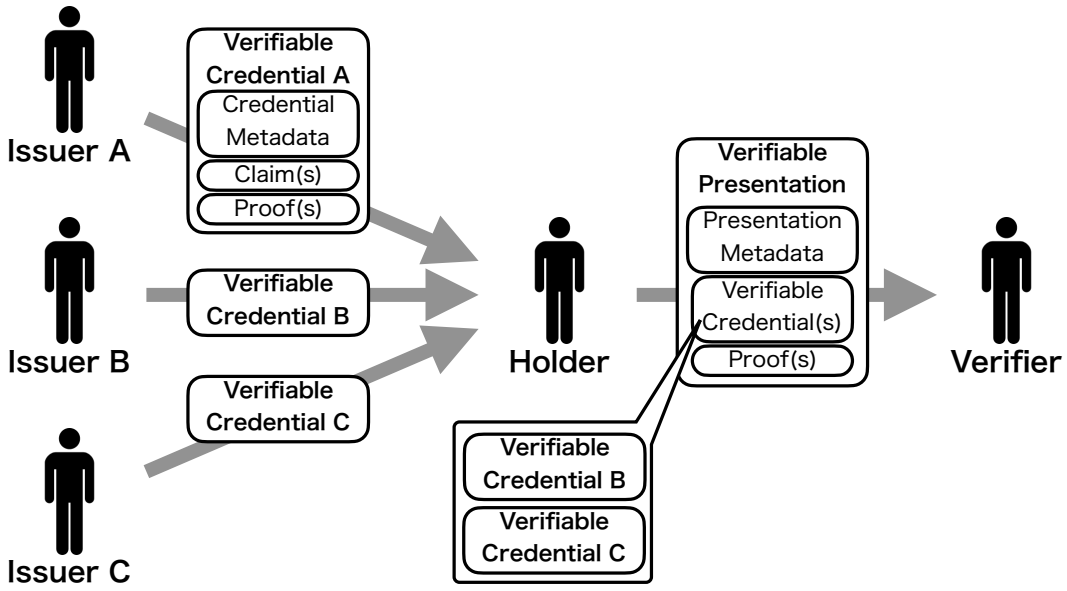

図 3.3　Verifiable Credentials のデータモデルと利用の例: 所有者が複数の Verifiable Credentials を提示する際は，単一の Verifiable Presentation 中に提示する Verifiable Credentials を含めて提示できる．

data 中に示される証明書発行者の識別子から，当該識別子に紐づく検証鍵を取得し，デジタル署名の検証に用いる．

Verifiable Credentials の標準では，証明書所有者が複数の Verifiable Credentials を提示するためのデータモデルである Verifiable Presentation を規定している． Verifiable Presentation は以下の要素から構成される．

- **Presentation Metadata**: Verifiable Presentation の提示者や，提示の目的などを示すメタデータ
- **Verifiable Credential(s)**: 提示される Verifiable Credentials
- **Proof**: Verifiable Presentation の完全性を示すための情報

図 3.3 に Verifiable Credentials のデータモデルと Verifiable Presentation を用いた複数証明書の提示フローを示す．

Verifiable Credentials および Verifiable Presentation は典型的には署名付きデータとして表現される [50]．したがって，Verifiable Credentials の標準では，JSON として表現する方法と，JSON のスキーマの定義が可能な JSON-LD を用いて表現する方法が例示されている．JSON で表現した場合，Proof が付与されたフォーマットとして JWT を用いて表現可能である [51]．

3.2.2 デジタル証明書の選択的開示

Verifiable Credentials などで表現されるデジタル証明書には，典型的には対象者にまつわる複数の情報が含まれ，その中には対象者のプライバシに関わる情報を含む場合がある．そのため，証明書の提示時には，提示の場面に応じて検証者が必要とする最小限の情報のみを提示することが好ましい．そこで，デジタル証明書の完全性を検証可能な状態を維持しながら，全体を開示することなく一部のみを開示する “選択的開示 (Selective Disclosure)” と呼ばれる仕組みが議論されている [25, 24]．選択的開示を適用することにより，単一の証明書が示す情報の集合を，当該集合のまま提示するのみならず，各要素に分解しながら提示できる．

選択的開示の具体的な手法の一例として，JWT をベースとした SD-JWT (Selective Disclosure for JWTs) が議論されている [25]．図 3.4 に SD-JWT の概要を示す．SD-JWT では，JWT のペイロードには直接情報を記載せずに，フィールド名，フィールド名に対応する値，salt 値の 3 要素を含む配列の暗号学的ハッシュ値を記載する．フィールド名，その対応する値，salt 値を含む配列は “Disclosure” と呼ばれる．すなわち，SD-JWT は，選択的開示が可能なフィールドに関して，該当フィールドに対応する Disclosure の暗号学的ハッシュ値のみが記載された JWT である．暗号学的ハッシュ値は不可逆であるため，JWT 単体では，選択的開示可能なフィールドは検証者に対して秘匿される一方で，通常の JWT と同様にその署名を検証可能である．当該フィールドの値を開示する際は，対応する Disclosure を検証者に提示する．検証者は，提示された Disclosure の暗号学的ハッシュ値を計算した上で，JWT が当該ハッシュ値を含むことを確認する．当該ハッシュ値を含むことが確認できれば，当該 Disclosure の示すフィールドと値が当該 JWT に含まれていると解釈できる．

3.3 ブロックチェーン技術

本節では，自律分散したノード間で協調しながら改ざん困難な台帳を構成する技術であるブロックチェーン技術を概説する．また，ブロックチェーン技術を用いたアプリケーションを構成するスマートコントラクトについても述べる．

3.3.1 ブロックチェーン技術の概要

ブロックチェーンは，仮想通貨 Bitcoin の基幹技術として発明された，特定の管理者なく動作する分散台帳技術である [52]．P2P ネットワークを構成する各ノードは，ネットワーク上に実装された仮想通貨の送金などを示すデータの履歴を全て保存する．典型的には，送金などを示す各データは署名付きデータとして表現される．各ノードが新規

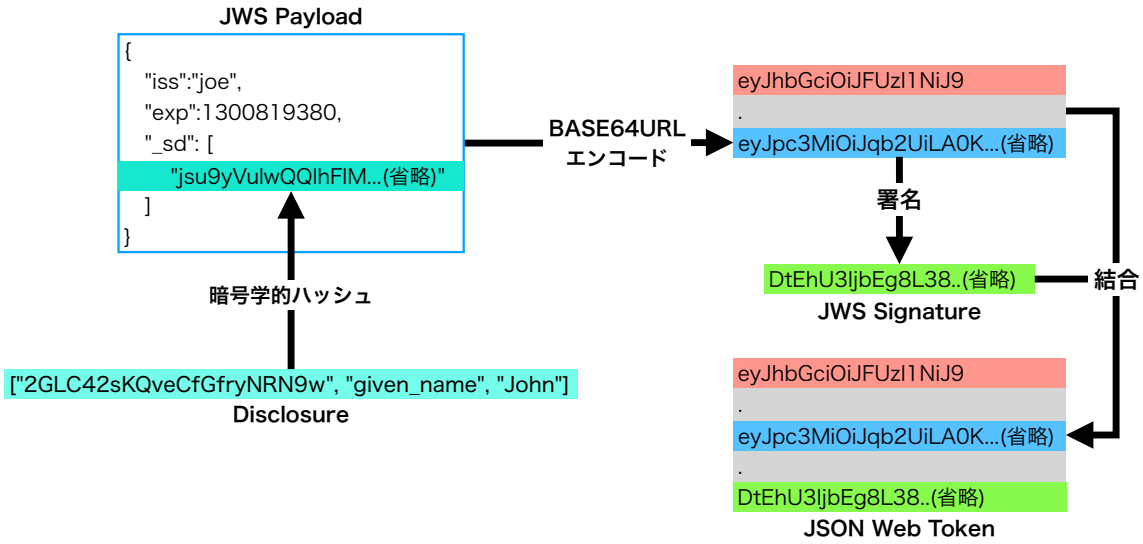

図 3.4　SD-JWT (Selective Disclosure for JWTs) の概要: JWT 自体には選択的開示を可能にする情報自体は含めず，フィールド名，その対応する値，salt 値を含む Disclosure の暗号学的ハッシュ値のみを含める．当該情報を開示する際は，JWT とともに開示する情報に対応する Disclosure を開示する．

データを受信した際は，各ノード上にそれぞれ保存された履歴を参照しながら当該データの署名とデータで示される操作を検証する．例えば，仮想通貨の送金する条件として “特定の検証鍵で検証できる署名の提示” が規定される場合，各ノードは “送金を示すデータが当該検証鍵で検証可能な署名が付与されていること” を検証する．検証に成功すると，各ノードは他のノードへ当該データをブロードキャストする．全てのノードが同様に新規データの検証とブロードキャストを行うため，結果的に P2P ネットワーク上の全てのノードが当該データを受信する．任意のノードは複数の検証済みデータを含む “ブロック” を作成し，ブロードキャストする．新規のブロックも同様に受信ノードにより当該ブロック内のデータの検証と，他のノードへのブロードキャストが行われ，検証に成功すると各ノードの台帳上へ保存する．Proof-of-Work などのコンセンサスアルゴリズムによって，全ノードで各ブロックのコンセンサスを形成するため，結果的に全てのノードは同じ台帳を持つ．ブロックのデータ構造とコンセンサスアルゴリズムにより，各ノードが持つ台帳の改ざんは困難である．また，最初のブロックは各ノードの実装上にハードコードされている．各ブロックは最初のブロックからのシーケンス番号 (ブロック高, Block Height) を持つ．したがって，特定のブロック内のデータ同士は，各データが含まれるブロックのブロック高によって相対時間を定義できる．

3.3.2 Ethereum とスマートコントラクト

Ethereum は，ブロックチェーンをベースとしたアプリケーションプラットフォームである [53]．各ノードは，チューリング完全なプログラムとその呼び出しを台帳上に記録する．各ノードは，改ざん困難な形式でプログラムとその呼び出しを記録するため，特定の管理者なく動作するプログラムを実装できる．

Ethereum 上で実行されるプログラムを “スマートコントラクト” と呼ぶ．各ノードは，スマートコントラクトをノードの実装内部に存在する Ethereum Virtual Machine (EVM) と呼ばれる仮想マシン上で実行する．典型的には，スマートコントラクトは Solidity などの高級言語で実装され，EVM 上で実行可能なバイトコードへコンパイルされる．Solidity で実装されるスマートコントラクトは，オブジェクト指向プログラミングにおけるクラスの概念のように，関数と変数を含む．Solidity は継承の概念もサポートしており，関数のインターフェースのみを定義した抽象コントラクトを定義可能である．継承の概念により，抽象コントラクトで定義された関数の内部のロジックを，当該抽象コントラクトを継承した具体コントラクトで定義できる．

スマートコントラクトを実行するためには，まず当該スマートコントラクトをブロックチェーン上へ記録する．スマートコントラクトをブロックチェーン上に記録し，実行可能な状態にすることを “デプロイ” と呼ぶ．また，スマートコントラクトにはデプロイ時のみ実行される特殊な関数であるコンストラクタを定義できる．当該スマートコントラクト中の関数を実行する際は，当該スマートコントラクトへの参照，関数の指定および関数への引数を含むデータをブロックチェーン上へ記録する．当該データは署名付きデータで表現される．したがって，特定の署名鍵を用いて作成した署名付きデータによって，特定の関数を実行したことが確認可能である．各ユーザは検証鍵の暗号学的ハッシュ値を基にした “Ethereum Address” を識別子として識別される．また，各ノードは EVM 上にスマートコントラクトの状態を保存する．スマートコントラクトとその呼び出しが改ざん困難な形式で記録されるため，各スマートコントラクトの状態は，そのコードによって定義された形式でのみ遷移する．したがって，EVM 上に保存される状態も同様に改竄困難である．

Ethereum とスマートコントラクトを活用して，様々なアプリケーションが提案されている [54, 55, 56]．例えば，Ethereum では，そのプラットフォーム自体に Ether と呼ばれる仮想通貨が実装されているが，独立した通貨の発行および支払いの手続きをスマートコントラクトとして定義し，新たな仮想通貨を定義できる [57]．また，定義された複数の仮想通貨の交換を実現するプロトコルをスマートコントラクトとして実装する試みもある [58]．

3.3.3 ブロックチェーン技術の活用例: Fair Exchange プロトコルを活用したデータの送受信

本節は，ブロックチェーン技術の活用例として，2 者間におけるデータの送受信の否認不可能性を担保する暗号プロトコルである Fair Exchange について概説する [59, 60, 61]．楽観的なプロトコルでは，送信者と受信者を仲介する中間者が，受信者が当該データの受信を否認した場合，あるいは送信者が送信を否認した場合に仲裁することで，送受信の否認不可能性を担保する．しかし，中間者が誠実に仲裁することを前提とするため，中間者が単一障害点となる．そこで，ブロックチェーンとスマートコントラクトを活用することで，中間者の機能を特定の管理者へ依存せずに実現する試みがある [62, 60, 63]．これらの試みでは，送信されるデータあるいはその一部をブロックチェーンに書き込むことで，受信者が当該データを受信できたことを証明可能にする．

Goldfeder らは，Bitcoin を用いた Fair Exchange プロトコルを提案した [64]．Fair Exchange の効率の向上をはじめとして，データマーケットや IoT（Internet of Things）など，様々な分野でこれらの交換プロトコルの活用が議論されている [65, 66, 67, 68, 69]．また，スマートコントラクトに一時的に仮想通貨を預け入れることで，ゲーム理論を援用し，双方が誠実に振る舞うインセンティブを設計する議論がある [70, 71]．これらの試みでは，両者は取引の開始前にスマートコントラクトへ仮想通貨を預け入れ，Fair Exchange を用いてデータを送受信する．データの送受信が確認できると，スマートコントラクトは預かった仮想通貨を払い出す．送信者がデータを送信しなかった場合，スマートコントラクトは当該仮想通貨を送信者へ払い出さない．同様に，受信者が “送信者が送信していない” と主張したものの，スマートコントラクトはデータの送信を確認している場合，受信者へ仮想通貨を払い出さない．したがって，預け入れた仮想通貨の払い戻しを受けることが両者にとって誠実に振る舞うインセンティブとなる．

3.4 本章のまとめ

本章では，本研究で活用する要素技術について概説した．まず，データの完全性を保証するためのデジタル署名の概念を整理し，署名付きデータの抽象データモデルを概説した．また，署名付きデータの一例として，JSON Web Signature と JSON Web Token について概説した． 次に，デジタル証明書の一例として，Verifiable Credentials とそのデータモデルについて概説した．また，デジタル証明書の選択的開示を議論し，実現する手法の一例として Selective Disclosure for JWTs を概説した．最後に，本研究の議論の一部で活用するブロックチェーン技術についても述べ，スマートコントラクトとその活用を議論した．

次章では，2 章で議論したクレーム検証のモデルを計算機を含む情報システムで実現す

るための関数モデルと，その課題について議論する．

第 4 章

情報システムを用いたクレーム検証

本章では，2 章で議論したクレーム検証を，情報システムとして実現するための関数モデルを定義する．また，既存の関連議論について整理した上で，情報システムにおけるクレーム検証の特性を議論する．

4.1 クレーム検証の関数モデル

2.3.3 節で議論したように，クレーム検証は，検証者の検証ポリシが示す検証基準に適合するかどうかを判定する問題として整理できる．そこで，クレーム検証を計算機を用いた情報システムとして実現するために，クレームが検証基準に適合するかどうかを判定する処理のインターフェースとして，抽象的な関数のモデルを定義する．

図 4.1 にクレーム検証の関数モデルを示す．検証者は，検証ポリシでクレームを妥当とみなす検証基準，およびクレームの検証に用いる根拠情報群を定義する．主張者は，主張時には当該検証ポリシに定義された根拠情報群とクレームを提示することで，検証者にクレームを検証させる．検証ポリシは，クレームおよび根拠情報群を入力とし，検証結果を出力する `Validation` 関数を定義する．`Validation` 関数の出力結果が，クレームが検証ポリシに併せて妥当かどうかを示唆する．本関数モデルを用いて，情報システムとしてクレーム検証を実現することは，検証ポリシに沿って `Validation` 関数を検証器として実装することであると定義できる．

検証ポリシで定義される検証基準に基づき，関数の内部での処理内容は決定される．そのため，検証者のコンテキストを反映し，対象となるクレームを妥当とみなす基準を柔軟に検証器として実装可能である．例えば，“過去に主張者が一切の不正をしたことがない” というクレームを考えると，直接当該クレームを示唆する根拠情報が存在し得ない場合も存在する．この場合，検証者のコンテキストに基づき，一定の範囲内では不正の履歴が存在しないことや，繰り返し誠実に振る舞った履歴を根拠情報として，不正がないとみなす基準を設計し，検証器を構成できる．

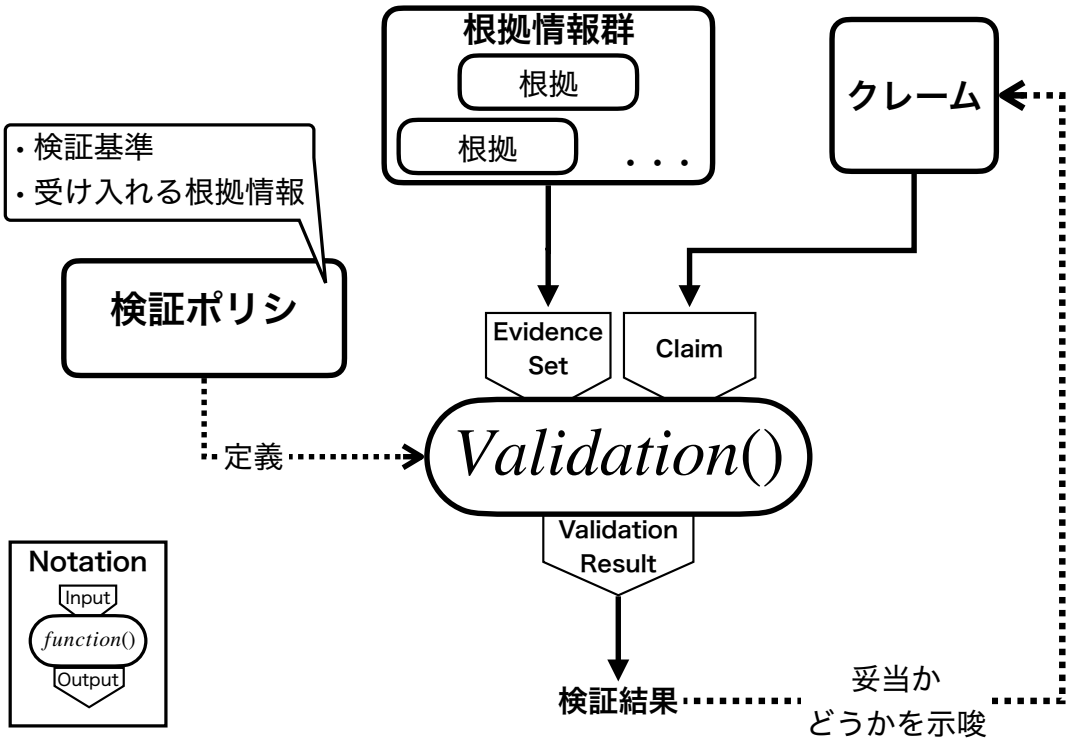

図 4.1　クレーム検証の関数モデル: 検証者は，検証ポリシの中でクレームの検証基準と，検証に用いる根拠情報群を定義する．検証ポリシは，クレームと根拠情報群を入力とし，検証結果を出力する `Validation` 関数を定義する．`Validation` 関数の出力結果が，クレームが検証ポリシに併せて妥当かどうかを示唆する．

4.2 既存研究とその課題

本節では，クレーム検証の関数モデルに対しての既存研究とその課題を議論する．何らかの対象を基準に基づき判定する問題は古くから研究されており，大別して，形式論理を基にした手法と，確率論を基にした手法へ分類される．

形式論理を基にしたシステムとして，専門家の判断を模倣することを目的としたエキスパートシステムが挙げられる [72, 73, 74, 75]．例えば，医療分野では病気の診断や治療法の選択などの，意思決定を支援するために広く活用されてきた [76, 77, 78]．エキスパートシステムは，様々な専門知識を必要とする判定問題に対して，判断の基準を蓄積した知識ベースと，推論エンジンによって構成される [73, 75, 79]．知識ベースに格納された知識を基に，推論エンジンで目的に対する推論を実施することで，判断を支援する [79]．推論エンジンは一階述語論理などで構成されることが多い．複数の基準から構成される推論は，その結果に至った要因が解釈困難になるケースが指摘されている [80]．本研究では，解釈可能性の議論と定義を礎に “人間が本質的かつ直感的に動作原理を理解

困難である程度” を “複雑さ” と定義する [81].

また，知識ベースを構成するためには，対象に対しての専門家へのヒアリングなどによって判断基準を見出す [82]．一方，専門家による判断は経験則に基づくヒューリスティクスである場合があり，知識ベース内での複数の基準が矛盾するケースがある [83].

一方，確率論を基にした議論としては，深層学習などを用いて対象の特徴量を抽出し，判定する手法が挙げられる．例えば，ソーシャルメディアにおける偽情報の判定を深層学習を用いる手法が議論されている [6, 4, 5]．偽情報の判定においては，対象の記事より，記事に含まれるクレームを抽出し，クレームの真偽を既存のデータセットを学習した機械学習モデルによって判定する．

確率論を基にした手法の課題として，判定の結果に不確実性があることが挙げられる．特に深層学習を用いた場合は，判定に至る過程がブラックボックスとなる [84, 8]．したがって，確率論による手法で結果を盲目的に受け入れることは懸念がある．また，特徴量を抽出するために大規模なデータの学習が必要であり，対象に関するデータセットが存在しない場合，システムを構成すること自体が困難である [7, 6].

4.3 情報システムとしてのクレーム検証の特性

本節では，情報システムで実現することを想定した際の，クレーム検証の特性を整理する．その上で，クレーム検証を実現するシステムを運用する際に想定される課題を議論する．

クレーム検証は，主張者が主張するクレームに対して，検証者が検証ポリシで定める検証基準に適合するかどうか判定する営みである．したがって，検証時点から見て，未来の事象を推論するタスクと比較して，現在あるいは過去の実績などを検証するタスクであると整理できる．また，クレーム検証の結果は，検証者の検証基準に適合するか否かであるため，特定のクレームに対して検証者の期待する結果は，適合あるいは非適合のどちらかに定まる．一方，未来の事象を推論するタスクは，その結果の正しさは当該事象が発生する未来において確定する．したがって，タスク実行時点ではその結果は一意に定まらないため，不確実性が許容されうる．検証者はアプリケーションの中で求められる確度に応じてクレームを検証し，検証済みのクレームを “ポリシに適合した” と捉えて後続の処理で活用する．この時，クレーム検証の結果に偽陽性あるいは偽陰性が生じる場合，後続の処理に影響を及ぼすと考えられる．したがって，クレーム検証において結果の不確実性は比較的許容されないと言えるだろう．

次に，検証ポリシで定める検証基準を検証器として実装した情報システムを，継続的に運用することを考察する．検証ポリシは，検証者の外的または内的要因によって変化しうる．例えば，外的要因としては，検証時に用いる根拠情報を取り巻く環境の変化が挙

げられる．具体的には，特定のクレームの検証時に，特定の団体が認定者として発行する証明書を根拠情報として用いていた場合，当該団体が解散・消滅することが考えられる．この時，すでに発行された証明書に関しても，当該団体の運用情報が存在しなくなることで，どのように証明書に示されるクレームを検証していたかが不明となるケースが考えられる．この場合，証明書の示すクレームの妥当性の判定が困難であることから，証明書の取り扱いを変更する必要がある．また，ポリシ変化の内的要因としては，アプリケーション全体の更改によってクレーム検証の目的が変化する可能性がある．この時，ポリシで規定されていた検証基準も変化することが考えられる．

特定の検証ポリシに沿って検証器が一度実装されたとしても，検証ポリシの変化が起こった場合，変化後のポリシと，過去のポリシに基づいた検証器の間で差異が生じる．この時，ポリシの変化に検証器が追従しなければ，クレーム検証の結果に偽陽性あるいは偽陰性が生じる可能性がある．したがって，検証器の実装にあたっては，実装後に検証基準が更改できる必要がある．本研究では，検証ポリシの変更に追従し，検証基準とその実装である検証器を更改できることを “更改可能性” と定義する．また，更改にあたっては，適切な更改が実施されたかどうか判断するために，更改前後の差異が明確であることが好ましい．

4.4 本章のまとめ

本章では，計算機を用いた情報システムでクレーム検証を実現するための関数モデルについて議論した．クレーム検証を検証者の検証ポリシが示す検証基準へ適合するかどうかの判定問題とみなし，検証ポリシによって `Validation` 関数が定義されると整理した．`Validation` 関数は，検証ポリシで規定される根拠情報群を入力とし，検証結果を出力する．情報システムとしてクレーム検証を実現することは，検証ポリシに沿って `Validation` 関数を判定器として実装することであると定義できる．判定器の実装にあたり，既存の手法として形式論理ベースの手法と，確率論ベースの手法が挙げられる．情報システムとしてのクレーム検証の特性として，現在あるいは過去の事象を検証するタスクであると整理でき，検証結果の不確実性が比較的許容されないタスクであると整理した．また，検証者の内的/外的両要因から検証ポリシは変化しうることから，検証器は更改可能であることが好ましいことを述べた．

次章では，本章で整理したクレーム検証の特性と既存研究の課題を受け，クレーム検証を情報システムで実装するためのモデルを提案する．

第 5 章

Shinken:
信頼に基づくクレーム検証モデル

本章では，前章で述べた特性と課題に対応するクレーム検証を実現するための，検証基準の構成手法である “Shinken モデル” を提案する．まず，Shinken モデルの要件を述べ，次に要件を満たすための課題を議論する．その後，提案手法である Shinken モデルの概要を述べる．

5.1 Shinken モデルの要件

本研究では，クレーム検証の特性に対応するため，検証基準を実装する検証器に以下の要件を定義する．

- **決定性**: 検証器の入力に対する検証結果が決定論的に一意に定まること
- **透明性**: 検証器の結果が偽陽性あるいは偽陰性である場合に，検証器の中から当該の結果に至った要因が特定できること

クレーム検証において，検証結果は検証者の検証ポリシで示される検証基準への適合あるいは非適合のどちらかである．したがって，結果の不確実性を排除するためには，検証器に入力するクレームと根拠情報群に対して，検証結果は決定論的に一意に定まることが好ましい．また，検証ポリシの変化などによって検証結果が偽陽性あるいは偽陰性であった場合，検証器を更改する必要がある．更改の際には，当該の結果に至った要因を特定し，当該部分の更改によって検証器をポリシに適合させることが可能であるべきである．

5.2 実現に向けた課題

本節では，要件を達成するための指針と，その実現に向けた課題を述べる．前節で述べた要件を満たすためには，検証器に実装される検証基準を形式論理に基づいて構成することが好ましい．しかし，形式論理に基づいてクレーム検証を実現するためには，エキスパートシステムと同様の課題を抱える．

検証器に実装される検証基準が複雑になることによって，検証基準の中に計算困難な問題を含む可能性がある．また，計算理論におけるライスの定理では，“非自明な特定の性質を満たすかどうかの判定は決定不可能である”とされる [85]．この時，“あらゆるクレームを検証できる”という性質が非自明な性質であることから，検証基準に含まれる命題が計算機で決定不可能なクレームが存在する．したがって，クレーム検証は計算機を用いた検証基準のみでは決定性を満たすことはできない．

5.3 提案手法

本節では，前節で述べた課題に対処しながら，クレームの検証基準を構成する手法である “Shinken モデル” を提案する[*1]．Shinken モデルでは，クレームの検証基準を，演繹的推論に基づいて定義する．その上で，検証基準の中で一部の命題を真であると仮定することによって，検証基準の決定性を担保する．ここで，真偽が確定できない命題に対して “真と期待する” とみなせることから，特定の命題に仮定を置くことを，クレーム検証における “信頼（trust）” と定義する．決定不可能な命題に対してのみならず，決定可能な命題であっても，クレーム検証のコンテキストに合わせて仮定できる部分に関しては信頼し，複雑な命題を省略することで，検証基準全体を簡略化できる．一方，信頼した仮定が破られた場合，検証結果に偽陽性あるいは偽陰性を生じる可能性がある．しかし，偽陽性あるいは偽陰性が生じた際は，信頼を導入した部分が要因であると推定できる．したがって，検証基準の中で信頼した部分を更改することで，検証器の改善が可能である．Shinken モデルを適用する際，クレーム検証のコンテキストに併せて偽陽性あるいは偽陰性の可能性を考慮しながら信頼する対象を検討し，検証基準を構成する必要がある．

5.4 Shinken モデルの適用例: 署名付きデータを用いたクレーム検証

本節では，Shinken モデルに基づき，検証器に実装される検証基準の構成例を議論する．例として，署名付きデータの解釈を取り上げ，信頼を導入することで決定性のある検証基準を構成できることを確認する．また，デジタル証明書の活用を題材に，Shinken

[*1] Shinken モデルの名称は，信頼 (“Shin”rai) に基づいて検証 (“ken”sho) することから名付けた．

モデルに基づく検証基準の更改可能性を検討する．

5.4.1 述語論理を用いた検証基準の構成

本節での議論における，述語論理を用いたクレームの検証基準の構成を概説する．Shinken モデルでは，クレームの検証を，以下のように定義する．

- 複数の述語によって構成される演繹的推論により，対象クレームが導出可能かどうか判定すること

検証者は検証基準として，対象クレームに至る演繹的推論を，形式的に妥当な形で構成する．その上で，“導出可能か判定すること” は以下のように定義できる．

- 対象クレームを結論とする演繹が形式的に妥当であり，健全であると判定すること

したがって，検証者は，形式的に妥当な演繹的推論を検証基準として検証ポリシに規定する．検証時には，検証器に対して根拠情報群を入力することで，演繹を構成する個々の述語が真であるかを確認する．個々の述語が真であることが確認できれば，演繹は健全であり，クレームの検証結果として，検証ポリシに応じて妥当であると判定できる．

5.4.2 信頼の導入による決定性のある検証基準の構成

本節では，Shinken モデルに基づき，信頼の導入によって決定性のある検証基準が構成可能であることを議論する．命題 S を，対象に関する任意のクレームとする時，デジタル署名を用いた S のクレーム検証を考察する．署名者と紐づいた鍵，署名，対象データを用いて，検証者は以下の真偽が判断できる．

- 署名時から対象データが完全性を保っていること
- 署名者が署名を付与したこと

本研究では，上記の真偽判断を “署名検証” と呼ぶ．“S が真ならば，署名者が対象に署名を付与する” という述語 I を考える．本研究では，述語 I における命題 S を “署名の意図” と呼ぶ．

図 5.1 に，デジタル署名を用いたクレームの検証基準である演繹的推論の概要を示す．署名者に紐づいた鍵で署名検証に成功する時，デジタル署名の特性により，署名者が署名を付与したことが推論できる．このとき，I が真であるならば，S も真であることが推論できる．このことから，署名が検証できれば，S が真であることの推論を構成できる．すなわち，署名者は署名の意図を定義することで，署名検証の結果から検証者が演繹的に推論可能な事象を規定できる．付録 A.2 に，本推論に基づく検証基準の Prolog による実

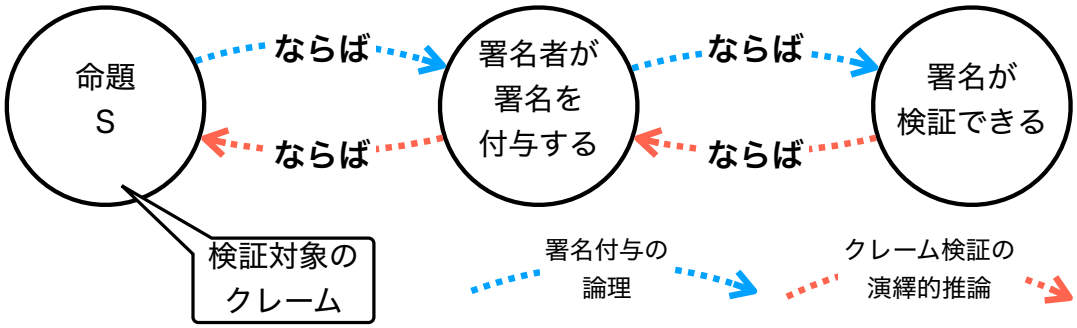

図 5.1　デジタル署名を用いたクレームの検証基準: 真の時に署名を付与する命題 S を署名の意図とする．このとき，デジタル署名の特性により，署名が検証できるならば，署名者が署名を付与したことを推論できる．また，署名者が意図に沿って署名することを信頼できれば，署名検証の結果から命題 S の真偽が演繹できる．

装を示す．

具体例として，特定のユーザに関する情報を，署名付きのデータを用いてクライアントに提供する認証プロトコルである OpenID Connect において，その署名と解釈について議論する [86]．OpenID Connect では，OpenID プロバイダが事前に検証した特定のユーザ (End-User) に関する情報を，署名付きデータである ID トークンに含めてクライアント (Relying Party) へ提示する．クレーム検証として整理すると，ID トークンに示される情報が検証対象であるクレームであり，ID トークンが根拠情報となる．また，ユーザが自身の情報をクライアントへ提供することを事前に許可した上で，OpenID プロバイダは ID トークンを発行する．つまり，ID トークンにおける署名の意図となる命題 S は，以下の 2 点である．

- ユーザが自身の情報の提示を事前に許可したこと
- ユーザの情報が OpenID プロバイダによって検証済みであること

この時，“ID トークンで示されるユーザの情報が OpenID プロバイダによって検証されており，ユーザが提供を許可したならば，OpenID プロバイダは ID トークンを発行する” ことが述語 I となる．したがって，クライアントは ID トークンの署名を検証することで，“ID トークンで示されるユーザの情報が OpenID プロバイダによって検証されており，ユーザが提供を許可したこと” を推論できる．これはプロトコルの定義によって署名の意図が規定されている例であると言えるだろう．

一方，署名者は S が偽である時にも署名を付与している場合，推論は健全ではない．すなわち，署名付与の論理である I 自体が真であることは，署名者の運用上の問題であり，計算機で決定不可能である．したがって，情報システムでクレーム検証を実現するためには，I が真であることを仮定し，信頼を導入することで署名検証の結果から S の真

偽の推論を健全なものとして構成できる．例えば，先に示した OpenID Connect の標準では ID トークンのやり取りの方式を定義しているが，OpenID プロバイダが事前に検証した情報を ID トークンに含めるかどうかは運用上の問題であり，プロトコルでは直接保証されていない．したがって，クライアントは OpenID プロバイダが正しく検証した情報を ID トークンに含めて発行すること仮定することで，ID トークンに示される情報を真であると推論する．

I が真である確度を高める方法として，規定された署名の意図を，事前に署名者と検証者の間で合意しておくことが考えられる．例えば法令などで規定されており違反すると罰則があるようなケースであれば，署名者の I が偽になるような挙動が，情報システムの外部から抑制されることが期待できる．法令によらずとも，特定のやり取りの形態として，事前に署名の意図をステークホルダ間で合意した上で，合意に沿ってやり取りが行われることを信頼する方法も考えられる．こうした合意や規則に基づき，クレームを検証する情報システム外のガバナンスによって，信頼を導入する点の妥当性を担保することが考えられる．

例えば，OpenID Connect では，典型的には OpenID プロバイダが特定のガバナンスの下，正しく検証した情報を ID トークンに含めて発行することを仮定する．一方，OpenID プロバイダに対する特定のガバナンスの外部に属する検証者が，クライアントとして ID トークンを検証するケースが考えられる．このようなケースにおいて，OpenID プロバイダが対象ユーザの情報を検証する際に用いた根拠情報などを ID トークンに含める拡張である OpenID Connect for Identity Assurance が標準化されている [87]. OpenID Connect for Identity Assurance で示される情報を用いて，ID トークンに含まれる情報を OpenID プロバイダが検証した方法と同様の方法で検証者は再度検証できる．例えば，日本国内において犯罪収益移転防止法に基づき，金融機関が運転免許証を用いて特定の主張者の本人確認を実施し，金融機関が OpenID プロバイダとして当該主張者の情報を示す ID トークンを発行するケースを考える [88]．この時，金融機関が特定の運転免許証を用いて本人確認を実施したことを示すために，ID トークンに運転免許証番号を含めて発行する．これによって，検証者は必要に応じて主張者に運転免許証の提示を求め，金融機関が確認した運転免許証と同一であることを確認することで，ID トークンに含まれる情報を金融機関が検証したことを確認できる．

5.4.3 デジタル証明書活用における更改可能性の検討

本節では，デジタル証明書の活用を対象に，Shinken モデルに基づいた検証基準の更改が可能であることを議論する．主張者のアイデンティティをクレームとして，検証者が検証するケースを議論する．主張者は，当該アイデンティティを示す証明書を保有して

いるとする．クレーム検証時には，主張者は検証者に対して根拠情報として当該証明書を提示する．この時，証明書を用いた検証基準は，前節で述べた署名の意図が “認定者が対象のクレームを検証した” という命題であると解釈できる．その上で，検証者はデジタル証明書に含まれる署名を検証することで，対象クレームを検証したとみなす．

図 5.2 に，デジタル証明書を用いた検証基準とその更改例を示す．前節の議論では，署名に用いる鍵が署名者に紐づいていることを前提に議論したが，攻撃などによって鍵が流出し，なりすました第三者が証明書を発行するケースが存在する [28, 29, 30]．その場合，認定者が証明書を発行したとは限らないため，署名の意図は推論できない．

この時，“認定者が証明書を発行する” という命題が真であるという仮定が破られている．したがって，検証基準に “鍵が認定者に紐づいてること” の確認を追加することで，検証器を改善できる．図 5.2 では，鍵とエンティティの紐付けを示す証明書である公開鍵証明書を用いて，認定者と鍵の紐付きを確認する検証基準を構成している．公開鍵証明書は “公開鍵証明書の発行者が鍵と特定エンティティの紐付きを確認したこと” を署名の意図とする証明書である．すなわち，検証者は，“公開鍵証明書の発行者が鍵とエンティティの紐付きを確認していること” を命題 S とした上で，“鍵とエンティティの紐付きが確認できたならば，証明書を発行すること” を述語 I として，真であると仮定する．当該仮定の下，公開鍵証明書の署名を検証することで，鍵とエンティティの紐付きを推論する．例えば，WebPKI における Domain Validated (DV) 証明書においては，“特定の FQDN を解決した先のホスト上で鍵を用いて署名作成できること” を確認する方法などが定められている [89]．この時，“特定の FQDN と鍵の紐付きを確認したこと” が具体的な署名の意図となる．また，Organization Validated (OV) 証明書においては，登記情報などの法的根拠に基づき組織の実在性を確認するなどの要件がある．このように，公開鍵証明書の署名の意図の具体として，発行者による紐付きの確認方法によって，公開鍵証明書に示される鍵とエンティティがどのように紐付くかが決定される．一方，公開鍵証明書に示される “発行者が鍵とエンティティの紐付きを確認した” こと自体も発行者の運用上の問題であり，真偽は決定不可能である．そのため，公開鍵証明書の発行者が自身のポリシに応じた確度で鍵とエンティティの紐付きを確認することを信頼し，認定者と鍵の紐付きを確認する検証基準を構成していると整理できる．より確度の高いエンティティと鍵の紐付きが必要とされる場合，OpenID Connect における OpenID Connect for Identity Assurance のような公開鍵証明書の発行者がどのように紐付きを確認しているか，を確認可能なスキームを導入することが考えられるだろう [86, 87]．WebPKI においては，公開鍵証明書の発行者は定期的な監査を受け，証明書を発行するにあたり適切な運用がされていることを監査される仕組みがある．これは，発行者が自身のポリシに沿って運用されていることを，外部的な監査とその結果の公開によって確認可能にしている，と整理できる．付録 A.3 に，本節の議論に基づく更改前後の検証基準の Prolog に

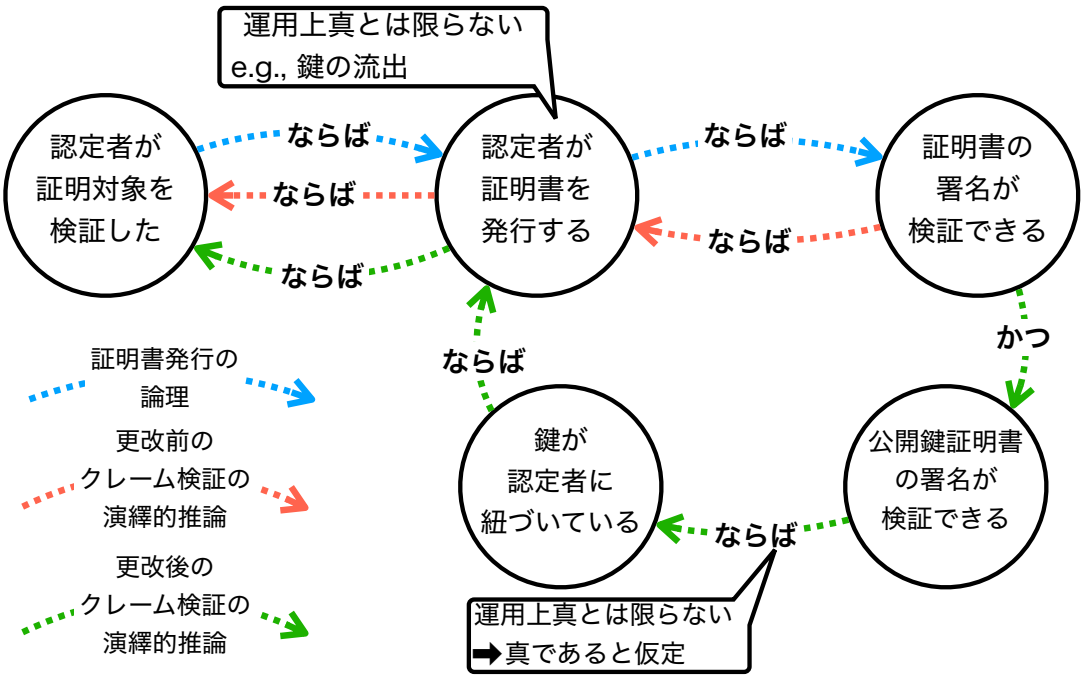

図 5.2　デジタル証明書を用いたクレームの検証基準とその更改例: 認定者と鍵の紐付きが仮定できない場合，署名検証の結果から “認定者が証明書を発行した” ことは演繹できない．本図では，クレームの検証基準を更改し，“検証鍵とエンティティの紐付けを確認したこと” を署名の意図とする公開鍵証明書を用いて，認定者と鍵の紐付きを検証する例を示す．

よる実装を示す．

5.5 本章のまとめ

本章では，情報システムとしてクレーム検証を実現するための Shinken モデルを提案した．情報システムとしてクレーム検証を実現するためには，検証ポリシの実装である検証器の決定性と透明性が要件であると整理した．要件を満たすためには検証基準を形式論理で構成することが好ましいが，計算機では決定不可能な問題が存在することを実現に向けた課題として述べた．そこで，検証基準に含まれる一部の命題を常に真であると仮定することを，クレーム検証における “信頼” であると整理し，信頼を導入することで決定性のある検証基準を構成する Shinken モデルを提案した．次に，Shinken モデルの適用例として，署名付きデータを用いたクレームの検証基準の構成例を示した．デジタル署名の検証が成功した際に推論できる命題を “署名の意図” と整理し，信頼を導入することで，デジタル署名を活用し，決定性のある検証基準を構成可能なことを示した．また，デジタル証明書の活用をデジタル署名の活用の一例と位置付け，決定性のある検証基準を構成したことにより，クレームの検証基準を更改可能であることを示した．

次章より続く 6 章と 7 章では，より具体的な事例における Shinken モデルの適用を議論する．6 章では，信頼に基づき検証基準を構成可能なことを，7 章では基準を更改可能なことをそれぞれ議論する．

第 6 章

ケーススタディ 1: 過去の商取引の結果検証

本章では，Shinken モデルを適用し，信頼を取り入れ，決定性のある検証基準を構成するケーススタディとして，商取引の結果検証を取り上げる．

6.1 背景: 商取引と Seller and Buyer's Dilemma

インターネット上で，デジタルデータの売買といった様々な商取引が行われている．商取引には，販売者 (Seller) からの商品の受け渡し，購入者 (Buyer) からの代金の支払いが含まれる．また，それぞれに関連した要件の合意のためのコミュニケーションなども含まれる．例えば，イラストなどのデジタルデータの制作依頼の取引においては，当該商品に関する要件の合意のためのコミュニケーションが取引中に実施されることが考えられる．このような商取引において，販売者および購入者にとって双方に相手が誠実に振る舞わないことによる経済的リスクがある．具体的には，先払いのケースでは購入者が代金を支払ったにもかかわらず，販売者が商品を受け渡さないケースが考えられる．後払いのケースにおいても，販売者が商品を受け渡したにもかかわらず，購入者が代金を支払わないケースが同様に考えられる．この問題は “Seller and Buyer's Dilemma” として知られている [70].

典型的には，この問題は第三者による仲介によって対処される．エスクロー (Escrow) と呼ばれる仕組みでは，取引開始時に購入者が仲介者に代金を預け，商品の受け渡しを当該仲介者が確認した上で，販売者へ代金を払い出す．また，商品の受け渡しが確認できない場合，仲介者は代金を購入者に払い戻す．仲介者を介して取引を成立させるには，仲介者が誠実に代金を預かり，商品の受け渡しを確認した上で払い出すことを販売者と購入者の双方が仮定しなければならない．したがって，そのような仮定を置ける仲介者が存在しない場合，仲介者を用いた取引は実施できない．特定の仲介者に依存せずに取引を実

施するために，特定の管理者なくシステムを動作させることが可能なブロックチェーン技術を用いて仲介者の機能を実現する試みがある [64, 70, 71, 65, 66, 67, 68]．これらの試みでは，特定のデータの送受信を保証する暗号プロトコルである “Fair Exchange” と，ブロックチェーン上で動作するプログラムである “スマートコントラクト” を活用する．

しかし，データの送受信の確認のみでは，先述の Seller and Buyer’s Dilemma への対処としては不十分である．例えば，制作依頼の取引において，販売者が商品を Fair Exchange を用いて受け渡しはしたが，当該商品が購入者の期待する質を満たさないケースが考えられる．このケースに対処するためには，商品が期待したものであると購入者が承認した際に報酬を払い出す仕組みが考えられる．しかし，購入者が商品の質に関わらず承認しない，あるいは取引開始時に販売者と合意した要件とは異なる要件を提示し承認しないケースが考えられる．これらの問題は，購入者の商品を承認する基準が曖昧であることから，商品が基準を満たしていることの確認が困難であることが要因である．一方，購入者の承認基準はスマートコントラクトで実装可能な基準であるとは限らない．したがって，Seller and Buyer’s Dilemma はスマートコントラクトのみでは解決できない．

購入者が商品を承認しない場合，紛争状態となり，購入者と販売者は紛争解決をする必要がある．通常，紛争解決はコストのかかる作業であるため，紛争に陥る傾向のあるエンティティを両者は取引開始前に取引相手として避けるモチベーションがある．したがって，取引開始前に対象エンティティが過去の取引を異常終了する傾向があることを確認できれば，紛争リスクを見積もることが可能である．リスクを見積もることで，取引相手として選択しない，あるいは代金を調節するなどの対処が可能であり，Seller and Buyer’s Dilemma が緩和できると考えられる．

6.2 クレーム検証の対象の整理と検証基準の構成

前節で議論したように，Seller and Buyer’s Dilemma を緩和するためには，各エンティティが “販売者あるいは購入者として関わった過去の取引の結果が正常終了したこと” をクレームとして，将来の取引相手が検証可能にすることが考えられる．本節では，過去の取引結果を検証可能にするための方向性と，Shinken モデルに基づいた検証基準の構成を議論する．

取引は，代金の支払いと商品の受け渡しを含む複数のタスクで構成される協調プロセスである．したがって，本ケーススタディでは，取引を示す協調プロセスのことを “取引プロセス” と呼ぶ．取引プロセスは，取引中に含まれるタスクと各タスクがどのエンティティに実行されるかで定義できるとする．この時，取引結果は，取引プロセスの定義に沿って取引が実行された結果である．したがって，取引プロセスの定義が存在し，それに沿ってプロセス中の各タスクの実行履歴があれば，取引結果が演繹的に推論できる．ま

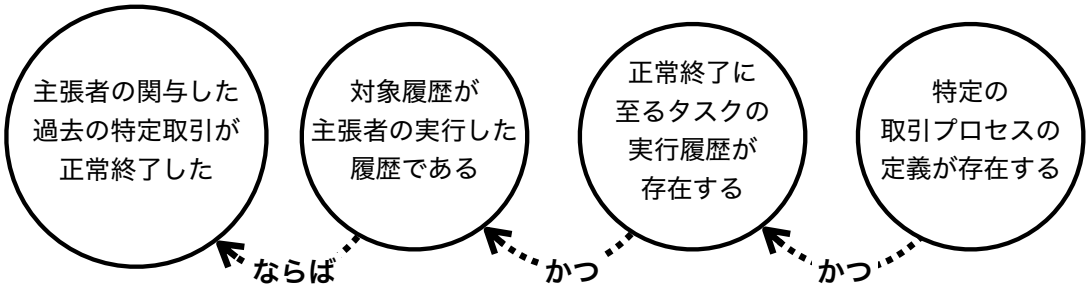

図 6.1 “特定の主張者の関与した過去の取引の結果が正常終了であった” ことの検証基準: 取引プロセスは，取引中に含まれるタスクと各タスクがどのエンティティに実行されるかで定義できるため，取引プロセスの定義と各タスクの実行履歴からその結果を演繹的に推論する．

た，各タスク実行履歴が特定のエンティティに紐づいていれば，当該エンティティが特定の取引プロセスに参加し，その結果が正常終了あるいは紛争を含む異常終了であったことを検証可能になる．以上から，特定の主張者の関与した過去の取引の結果が正常終了であった，というクレームの検証基準を図 6.1 に示す．なお，紛争を含む異常終了は，取引プロセスの定義中で各タスクの実行の結果による状態遷移を定義し，各タスクの実行履歴から取引の状態遷移を確認することで，同様の検証基準で検証できる．

ここまで議論した検証基準によって過去の取引結果を検証するには，取引の定義と当該定義に沿った履歴を検証者が取得可能にする必要がある．前節で議論した既存の試みでは，ブロックチェーン上に Fair Exchange を実行するタスクをスマートコントラクトとして定義する手法であると整理できる．この時，ブロックチェーン上で実行された取引の履歴は台帳上に記録され，ブロックチェーンの特性により高可用性を持ち，改ざん困難であることを検証者は仮定する．すなわち，履歴が存在することを検証するために，ブロックチェーンの動作および性質を仮定し，信頼することによってクレーム検証を実現できると整理できる．

図 6.2 に，2.3.1 節で示した基本モデルのやり取りに基づいた，商取引の結果をクレームとするクレーム検証のやり取りを示す．取引の参加者は主張者として，過去の取引の結果をクレームとして主張し，将来の取引相手は検証者として当該結果を検証する．将来の取引相手は先述の検証基準を自身の検証ポリシに定める．その上で，各取引はブロックチェーン上で実施され，履歴が記録される．記録された履歴は，当該取引の結果に対する根拠情報となる．この時，取引の参加者は当該履歴に対する参照を将来の取引相手に提示し，将来の取引相手は履歴を参照することで取引結果を検証できる．付録 A.4 に，本節の議論に基づいた取引の結果をクレームとする検証基準の Prolog による実装を示す．

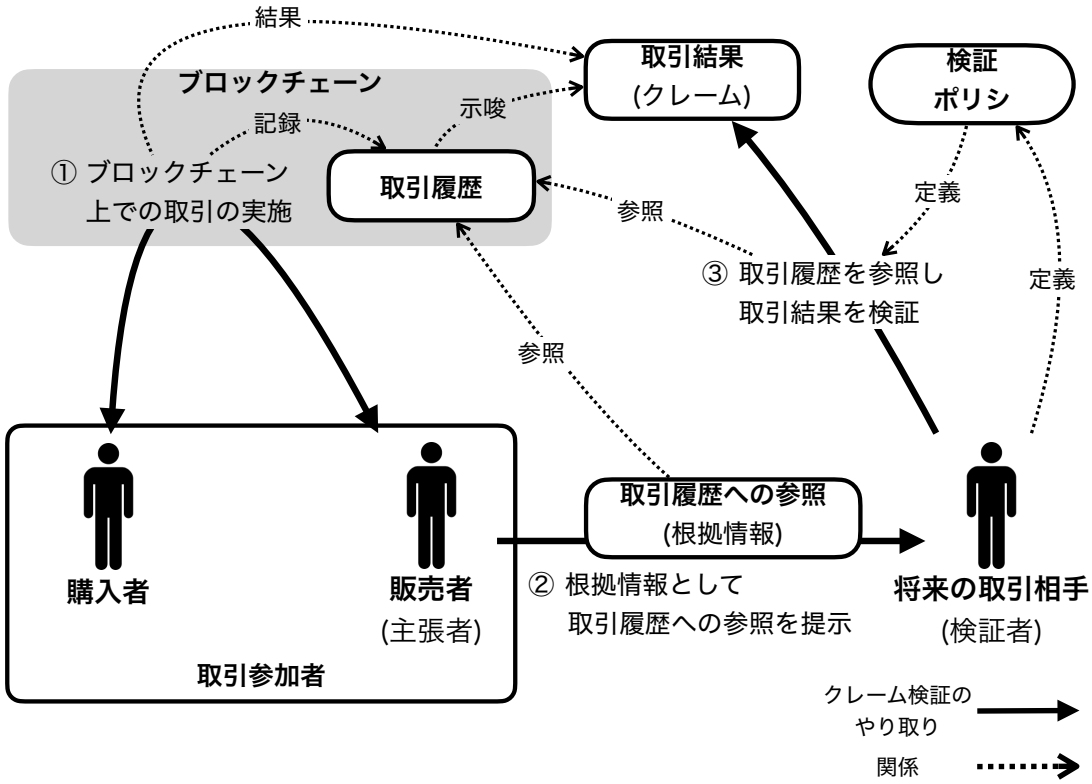

図 6.2　商取引の結果をクレームとするクレーム検証のやり取り: 本図では，販売者が主張者として，検証者である将来の取引相手に，自身の過去の取引結果をクレームとして主張する例を示す．将来の取引相手は，図 6.1 で示した検証基準を検証ポリシに定める．販売者は，過去の取引履歴への参照を根拠情報として提示する．将来の取引相手は取引履歴を参照し，検証基準に沿って過去の取引結果を検証する．

6.3 取引結果の検証を実現する取引フレームワークの設計

既存手法である Fair Exchange を用いたエスクローは，仲介者なく取引を実現することを主な目的としている．そのため，過去の取引の履歴を取得，およびその結果を検証できるインターフェースはデザインされていない．また，用いられる Fair Exchange の具体的な実装手法や，代金の支払いあるいは商品の受け渡し以外の様々なタスクが含まれることを考慮すると，様々な取引プロセスが考えられる．さらに，既存手法では，特定の Fair Exchange のプロトコルを含む取引プロセスがスマートコントラクトにハードコードされているため，特定の取引プロセスにしか対応できない．これらのことから，本ケーススタディでは様々な取引プロセスの結果の検証を実現するために，取引プロセスが定義可能であり，取引結果を検証可能な取引フレームワークを設計する．

6.3.1 要件定義

本節では，取引フレームワークの要件を定義する．まず，エスクローを実施するための要件として，以下を定義する．

- **ER-1**: 取引開始時点で，代金分の購入者の資金をスマートコントラクトに預け入れ，購入者が他の取引で払い出せないようにする
- **ER-2**: 預け入れた資金は，取引プロセス中の定義されたタイミングで販売者に払い出す
- **ER-2'**: 取引が異常終了した際は，預け入れた資金は購入者へ払い戻す

ER-1 は，販売者が商品を受け渡した際に，購入者が代金を支払う能力を保証する．**ER-2** は，販売者が商品を受け渡した際に，販売者が代金を受け取れることを保証する．**ER-2'** は，販売者が誠実に商品を受け渡すインセンティブを促す．

次に，取引履歴の閲覧のための要件を以下に定義する．

- **HR-1**: 特定のエンティティ間での取引履歴が確認可能である．
- **HR-2**: 履歴は取引が終了して以降，改竄できない．
- **HR-3**: 取引の終了状態が正常終了か，異常終了であることが確認可能である．
- **HR-3'**: 異常終了であった場合，紛争によるものか，どちらか一方のエンティティの責任によって異常終了に至ったかを確認可能である．

HR-1 により，特定のエンティティの過去の取引の履歴を取得可能なことが保証される．**HR-2** により，特定のエンティティの取引終了状態の傾向を偽ることを困難にする．**HR-3** により，特定のエンティティが取引を異常終了する傾向にあることを確認可能にする．**HR-3'** により，特定のエンティティが異常終了の原因となる行動をするか，あるいは紛争を引き起こす傾向にあることが確認可能になる．

次に，上記の要件を満たした取引プロセスを定義可能なフレームワークの要件として，以下を定義する．

- **FR-1**: “取引プロセス中に含まれるタスク” が取引プロセス毎に定義可能である．
- **FR-2**: “取引の状態” が取引プロセス毎に定義可能である．
- **FR-3**: 取引の状態の遷移は，“定義されたタスクの完了” によって遷移する．

FR-1 は，取引プロセスに含まれる必要なタスクを定義可能にする．**FR-2** は，取引プロセスの定義に沿った取引の状態を定義し閲覧可能にする．**FR-3** は，過去の取引の状態がその取引プロセスの定義に沿って示されていることを保証する．

6.3.2 前提

本節では，取引フレームワークの設計における前提を述べる．本取引フレームワークはブロックチェーン上に実装され，Ethereum のようにスマートコントラクトが実装可能なブロックチェーンを前提とする．また，Shinken モデルに基づき，3.3 節で述べたブロックチェーン自体の動作および特性は保証されていることを仮定する．したがって，ブロックチェーン上で実装されたスマートコントラクトは実装された通りに動作する．それぞれのエンティティは Ethereum における Ethereum Address などのブロックチェーン上における識別子で識別する．代金の支払いは，Ethereum における Ether のような，ブロックチェーンにハードコードされている仮想通貨で支払う．本取引フレームワークでは，取引プロセスは分岐のない直線的なプロセスを想定する．したがって，取引の終了状態は，正常終了，タスクの期限切れ (タイムアウト) による異常終了，あるいはその他の理由による異常終了の 3 パターンとなる．

6.3.3 取引フレームワークの概要

図 6.3 に取引フレームワークの概要を示す．本フレームワークをベースとして，取引プロセスの定義を含み，“EHR コントラクト (Escrow and History Repository contract)” を実装できる．各取引は，EHR コントラクト上で実施される．取引の実施にあたり，EHR コントラクトはブロックチェーン上にデプロイされており，取引が開始可能な状態にあるとする．

EHR コントラクトのフレームワークとして，以下のインターフェースを定義する．

- **Init Escrow Interface**: EHR コントラクト中で定義される取引プロセスに応じたエスクローを開始するためのインターフェース
- **Task Execution Interface**: EHR コントラクト中で定義される取引プロセスに含まれるタスクを実行するためのインターフェース
- **History Viewer Interface**: EHR コントラクト上で実行された取引履歴を閲覧するためのインターフェース

EHR コントラクトは，取引プロセスの定義として以下の項目を含む．

- **Task Validator(s)**: 取引プロセス中に含まれるタスクの完了を検証する関数 (群)
- **Task Sequence**: 取引プロセス中に含まれるタスクの順序を示す, Task Validator の配列
- **Timing of Payment**: Task Sequence 中の特定タスクの終了時点における販売

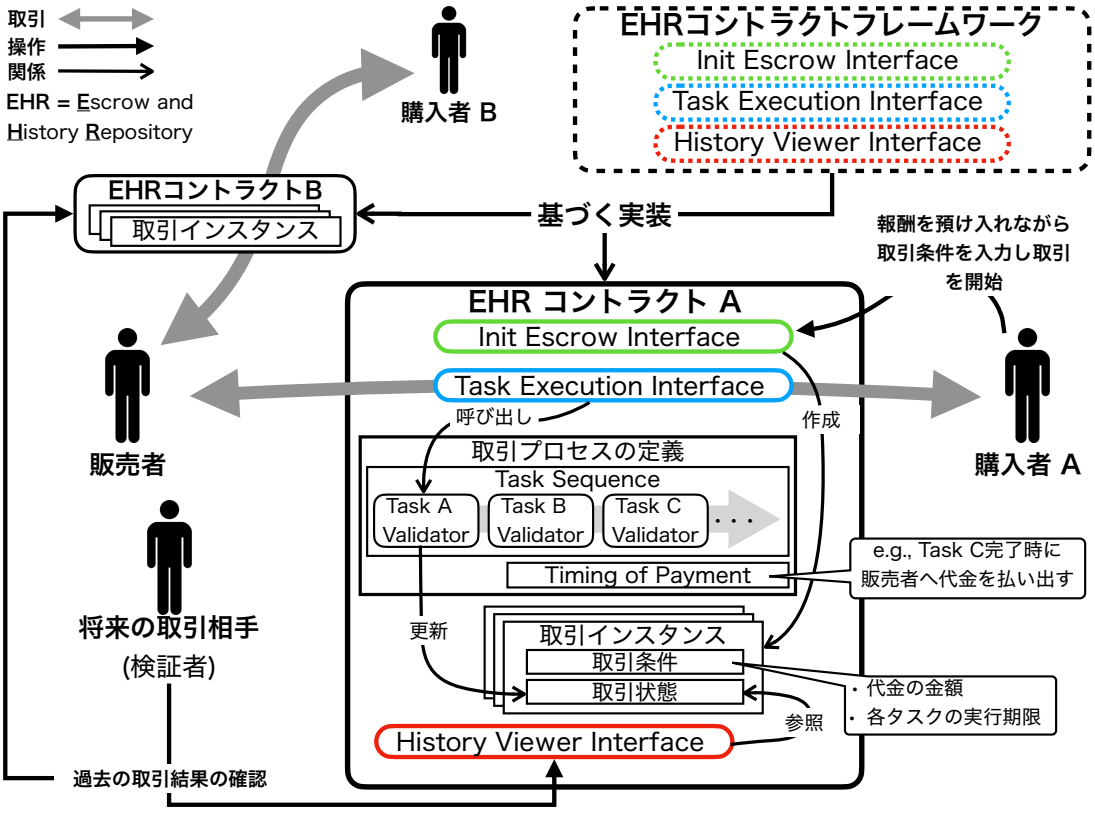

図 6.3　取引フレームワークの概要: 本フレームワークは (1) エスクローの開始，(2) 取引プロセス中のタスクの実行，(3) 履歴の閲覧，の 3 つのインターフェースを定義する．本フレームワークを利用した EHR コントラクトは，取引プロセスの定義を含む．本図は，将来の取引相手が検証者として，本フレームワークを利用した様々な取引プロセスの履歴を検証可能であることを示している．

者への代金払い出しの定義

特定の EHR コントラクト上で取引を開始する際は，購入者は以下の項目を取引条件として定義し，Init Escrow Interface へ入力する．

- 代金の金額
- Task Sequence 中の各タスクの実行期限

同時に，購入者は代金の合計額を EHR コントラクトへ預け入れる．入力を受け，EHR コントラクトは，定義された取引プロセスに基づく個別の取引を示す “取引インスタンス” を作成する．取引インスタンスは取引条件と，取引の状態を示す値を含む．次に，購入者と販売者は，各 Task Validator で定義されるタスクを実行し，Task Execution Interface へタスクの実行を入力する． Task Execution Interface は，取引インスタンスの状態に応じて，Task Sequence で示される Tack Validator によって各タスクの完了を

検証する．また，検証の結果によって，取引インスタンスの状態が遷移する． Timing of Payment で定義されたタスクが完了した時，EHR コントラクトは条件で示された金額の代金を販売者へ払い出す．

取引インスタンスの作成以降，将来の取引相手は History Viewer Interface を通じて，各取引インスタンスの状態を確認可能である．したがって，将来の取引相手は特定の販売者あるいは購入者の関与した取引の状態を確認可能である．また，本フレームワークに基づく EHR コントラクトであれば，取引プロセスの定義が異なったとしても，同様に History Viewer Interface から取引の状態を閲覧可能である．

6.3.4 取引の状態遷移と各終了状態に至る責任

本節では，本フレームワークを用いた取引における取引の状態遷移を定義する．図 6.4 に取引の状態遷移の例を示す．本フレームワークでは，取引プロセス中の各状態を，特定のタスクの実行待ち状態と定義する．取引条件で示された実行期限内に特定のタスクが実行されなかった場合，当該タスクは期限切れとみなす．したがって，その場合は，特定のタスクがタイムアウトしたことによって，当該取引が異常終了したと解釈する．

各 Task Validator は，タイムアウト以外の理由によって取引を異常終了できる．この場合，Task Validator は取引インスタンス中の取引の状態を，特定の異常終了状態へ遷移させる．EHR コントラクトにより定義される取引プロセス毎に，異常終了のパターンは異なる．したがって，特定のタスク実行待ち状態から複数の異常終了状態に遷移する場合がある．

取引が異常終了した際，将来の取引相手は販売者か購入者どちらの責任によって当該異常終了に至ったかを確認するモチベーションがある．特定のタスクがタイムアウトした場合，当該タスクが割り当てられていたエンティティに異常終了に至った責任がある．タイムアウト以外の異常終了のケースでは，それぞれの取引プロセスの定義に応じて各異常終了状態に至った責任が，販売者あるいは購入者のどちらにあるか明確化することで，責任があるエンティティを定義できる．

6.3.5 正常終了を促す両エンティティのインセンティブ設計

本節では，代金の支払いタイミングの定義により，販売者と購入者の双方に取引の正常終了を促すインセンティブを設計する．本フレームワークでは，購入者は取引の開始時に代金を EHR コントラクトへ預け入れる．購入者は代金を支払い，商品を入手しようとすることを考えると，購入者は代金よりも商品に価値があると考えているとみなせる．したがって，商品を得るため，購入者はタスクを正常に完了するインセンティブがある．次に，販売者は代金を得て商品を受け渡すため，販売者は商品よりも代金に価値がある

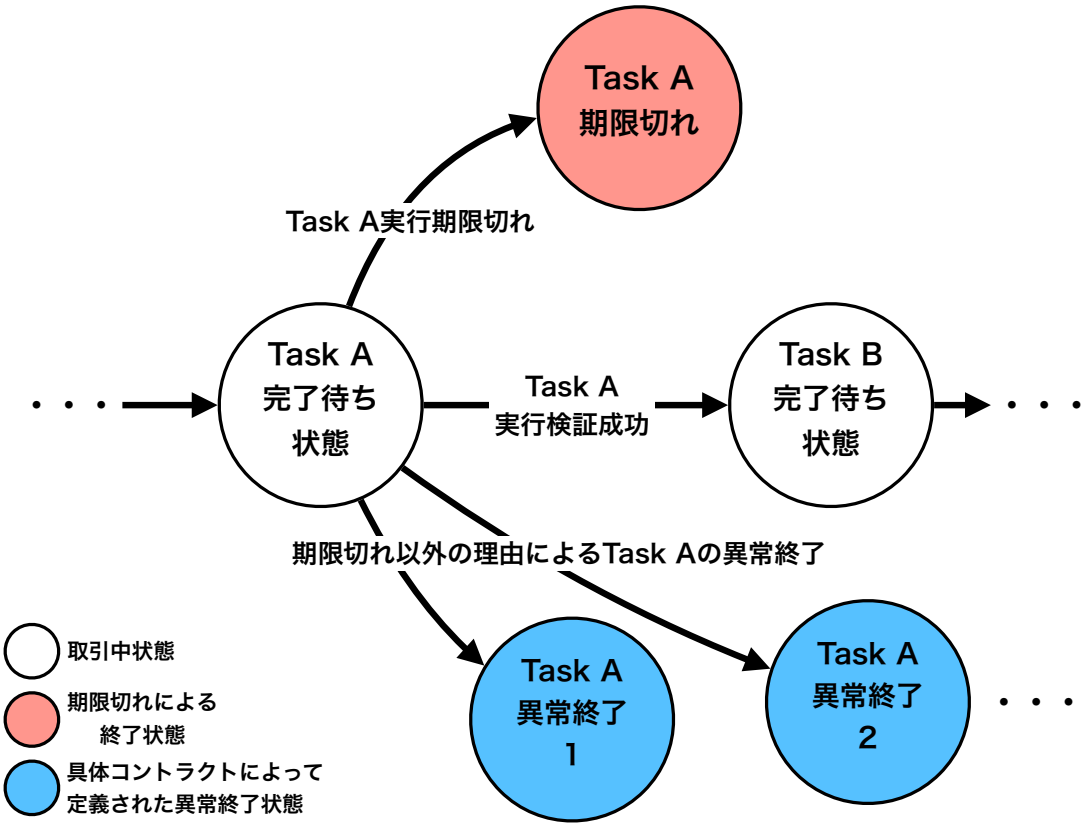

図 6.4　取引の状態遷移: 取引プロセスの状態は特定のタスクの実行待ち状態である．取引条件で示された実行期限内に特定のタスクが実行されなかった場合，当該タスクは期限切れとみなす．各 Task Validator は，タイムアウト以外の理由によって取引を異常終了できる．

と考えているとみなせる．したがって，購入者同様に，代金を受け取るため，販売者は EHR コントラクト上に定義された受け渡しのタスクを完了するインセンティブがある．

また，将来の取引相手が各取引の状態を確認可能であるため，販売者と購入者は，割り当てられたタスクがタイムアウトしていた場合，当該取引の異常終了の責任があるとみなされるリスクがある．したがって，両者はそれぞれのタスクがタイムアウトする前に，誠実にタスクを完了するインセンティブがある．さらに，EHR コントラクト上で定義された異常終了においても，同様にその責任があるとみなされるリスクがある．以上より，商品と代金の交換を EHR コントラクト上で実施し，取引の終了状態を確認可能にすることで，両者には誠実に振る舞い，取引を正常終了させるインセンティブがある．

6.4 実装

本節では，本フレームワークのプロトタイプ実装について述べる．本フレームワークを Solidity を用いて抽象コントラクトとして実装し，Ethereum の仮想環境上で実行し

表 6.1　TransactionDefinition Structure

Field	Solidity における データ型	概要
taskSequence	array(string)	取引プロセスに含まれる タスクの順序を示す Task Validator 関数名の配列
taskPointer	mapping(string → function)	Task Validator 関数名と Task Validator 関数自体を 対応づけるマップ
paymentTiming	array(uint)	代金を購入者に払い出す タイミングを示す配列. 各要素は *taskSequence* フィールドに示される タスクのインデックスを示す.

た．本フレームワークを用いた EHR コントラクトは，当該抽象コントラクトを継承した具体コントラクトとして実装できる．本実装では，エンティティの識別子は Ethereum Address であるとし，代金の支払いは Ether 立てで支払われるとする．

6.4.1　データ構造

本実装では，以下のデータ構造を抽象コントラクト中に定義する．

- **TransactionDefinition Structure**: 取引プロセスの定義を示すデータ構造
- **TransactionCondition Structure**: 個別の取引の条件を示すデータ構造
- **EscrowInfo Structure**: 個別の取引の全体の情報を示すデータ構造

TransactionDefinition Structure

表 6.1 に取引プロセスの定義を表す *TransactionDefinition* structure の概要を示す．抽象コントラクトでは，本データ構造は定義されるのみで，具体コントラクトの中でそれぞれの値を代入することで取引プロセスを定義する．

TransactionCondition Structure

表 6.2 に取引条件を表す *TransactionCondition* structure を示す．*duration* フィールドでは，直前のタスク実行からの，対象タスクの実行までのタイムリミットをブロック

表 6.2　TransactionCondition Structure

Field	Solidity における データ型	概要
seller	address	販売者の Ethereum address
buyer	address	購入者の Ethereum address
reward	array(uint)	各支払いタイミングにおける代金の 金額の配列．各要素のインデックスは， $TransactionDefinition.paymentTiming$ フィールドに示される支払いタイミングの インデックスを示す．
duration	array(uint)	各タスクのタイムアウトするまでの期限を 示す値の配列．各要素のインデックスは， $TransactionDefinition.taskSequence$ フィールドに示される Task Validator 関数名の インデックスを示す．
nonce	bytes32	各取引インスタンスを識別可能にするための乱数

高で示す．例えば，第一のタスクがブロック高 10 の時にブロックチェーンへ記録され，$duration[0] = 10$ の時，次のタスクを実行するエンティティはブロック高 20 までに当該タスクを実行しなければならない．ブロック高 20 以降に実行されていなかった場合は，当該タスクはタイムアウトする．

それぞれの取引インスタンスは *TransactionCondition* structure の暗号学的ハッシュ値である識別子 *EscrowId* で識別される．同じ条件の取引を繰り返し実行可能にするため，*TransactionCondition* structure は乱数である *nonce* を含む．

EscrowInfo Structure

表 6.3 に，特定取引の全体の情報を表す *EscrowInfo* structure を示す．購入者が取引を特定の条件のもと開始する際は，EHR コントラクトは *EscrowInfo* structure のインスタンスを作成する．EHR コントラクトは *EscrowInfo* 内の *condition.duration* フィールドと *history* フィールドを用いて，特定のタスクがタイムアウトしたかどうかを判断する．*state* フィールドは取引インスタンスの状態を示す．取引の状態を表す状態コードの詳細は 6.4.3 節で述べる．

表 6.3　EscrowInfo Structure

Field	Solidity における データ型	概要
condition	TransactionCondition	購入者によって提示される取引条件
state	uint	取引の状態コード
history	array(uint)	各タスクが実行された際のブロック高. 各要素のインデックスは, $TransactionDefinition.taskSequence$ フィールドに示される Task Validator 関数名の インデックスを示す.

6.4.2　フレームワークの実装

図 6.5 に実装の概要を示す．フレームワークとして，抽象コントラクトに以下の関数を定義する．

- **InitEscrow function**: 取引条件を記録し，取引インスタンスを作成する関数
- **processTask function**: タスクを実行するための関数．各エンティティが当該関数を実行すると，EHR コントラクトは対応する Task Validator 関数を呼び出し，その結果に応じて取引の状態を遷移させる．
- **getHistoryAsSeller/getHistoryAsBuyer function**: 過去の取引履歴を閲覧するための関数

抽象コントラクトは，空の *TransactionDefinition* structure のインスタンスを定義する．取引プロセスを定義する EHR コントラクトは，抽象コントラクトを継承することで実装できる．EHR コントラクトは，各タスクの完了を確認する関数を Task Validator として定義する． 各 Task Validator 関数は，対応するタスクを実行する責任のある販売者あるいは購入者を指定する．EHR コントラクトのコンストラクタでは，*transactionDefinition* の各フィールドを定義する．また，EHR コントラクトは複数のタスクを含む取引プロセスを定義する．例えば，データを送信するための Fair Exchange の各ステップを，一連の Task Validator の集合として定義できる．

定義された取引プロセスを実施する際は，購入者は抽象コントラクトで定義されている `initEscrow` 関数を呼び出す．`initEscrow` 関数は取引インスタンスを *EscrowInfo* structure で表現し，記録する．この時，購入者は代金の合計額を EHR コントラクトに預け入れ，取引条件を引数として入力する．引数には以下の項目が含まれる．

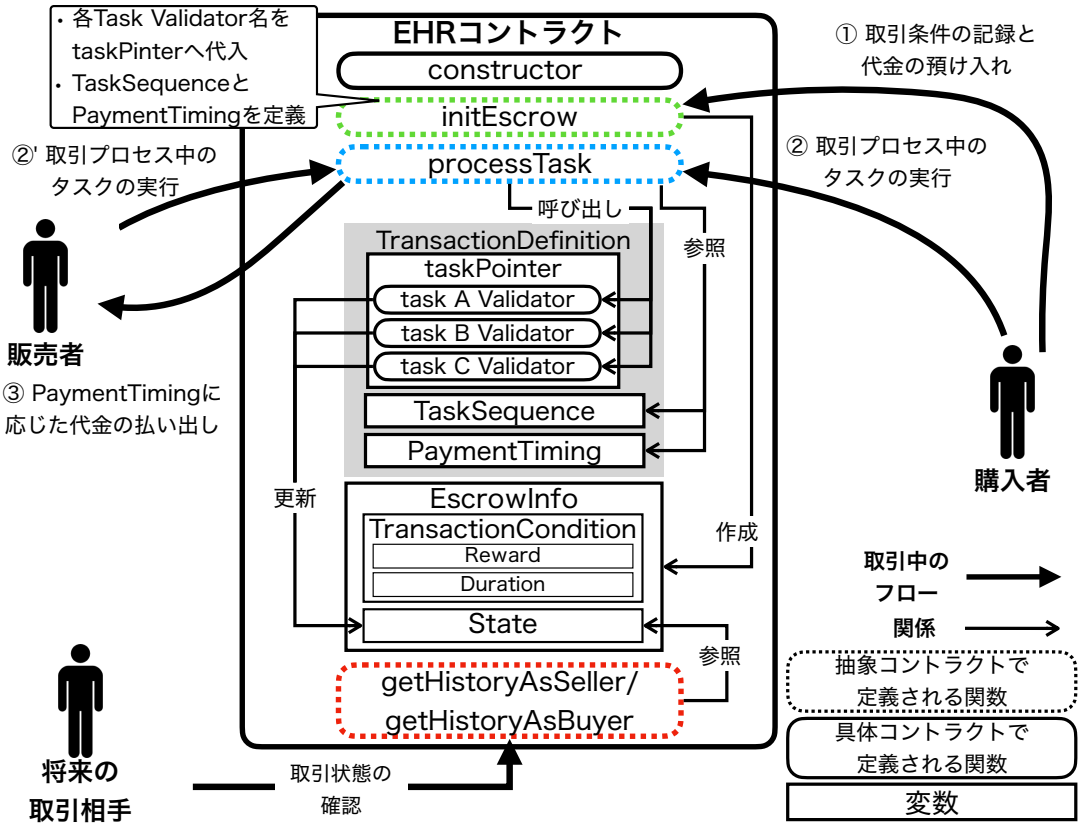

図 6.5　取引フレームワーク実装の概要: フレームワークを抽象コントラクトとして実装し，継承した具体コントラクトである EHR コントラクトが個別の取引プロセスを定義する．

- 販売者の識別子
- 各支払いタイミングにおける代金の額
- 取引中に含まれるタスクのタイムリミット

EHR コントラクトは，引数として受け取ったそれぞれの値を *TransactionCondition* structure の値へ代入する．また，引数で示されない `nonce` はランダム値を，`buyer` は `initEscrow` を実行したエンティティの Ethereum Address を EHR コントラクトが代入する．取引インスタンスは取引条件と当該インスタンスの状態を含む．取引インスタンスは，取引条件の暗号学的ハッシュである *EscrowId* で識別される．以降，特定の取引インスタンスに対して操作する際は，*EscrowId* を指定して各関数を実行する．取引インスタンスに対して，タスクを実行する際は，各エンティティは `processTask` 関数を実行する．`processTask` 関数は，引数として実行されるべき Task Validator 関数名と，タスクの完了を確認するために必要なデータを引数として受けとる．その上で，`processTask` 関数は，*transactionDefinition.taskSequence* フィールドに示される特定のタスクに対する Task Validator 関数を実行する．`processTask` 関数は，取引インスタン

スの状態が指定されたタスクを実行するべき状態であること，当該タスクを実行する責任を持つエンティティが実行していることを確認する．上記 2 点の確認が取れると，Task Validator はタスクの完了を確認する． Task Validator がタスクの完了を確認できると，`processTask` 関数は当該取引インスタンスの状態を次の状態へと遷移させる．この時当該タスクが，*transactionDefinition.paymentTiming* で示される完了とともに支払いを伴うタスクであった場合，EHR コントラクトは *transactionCondition.reward* に示された金額の代金を販売者へ払い出す．それぞれのタスクが *transactionCondition.duration* で示された期限内に実行されなかった場合，当該取引インスタンスの状態は，当該タスクがタイムアウトしたとして異常終了状態となる．

取引が完了したのち，将来の取引相手は当該取引インスタンスの状態を `getHistoryAsSeller` 関数または `getHistoryAsBuyer` 関数を通じて確認できる．両関数は，エンティティの識別子を引数としてとり，当該識別子に紐づけられた *escrowInfo* の集合を返す．したがって，各エンティティの将来の取引相手は，新規の取引を開始する前に当該エンティティが関与した過去の取引インスタンスの状態を確認できる．これによって，特定のエンティティが特定 EHR コントラクト上での取引において取引を異常終了，あるいは正常終了させたことを確認できる．

6.4.3 状態コードの定義

本節では，実装における各取引インスタンスの状態を示す状態コードを概説する．図 6.6 に，実装における状態遷移と状態コードを示す．本ケーススタディでは，取引は直線的なプロセスを想定しているため，取引プロセス中の状態コードは 100 番台であると定義する．全てのタスクが正常に完了し，取引プロセスが正常終了した場合の状態コードは 200 とする．したがって，本実装においては取引プロセスは最大 100 タスクで構成される．各タスクがタイムアウトした場合，取引プロセス中の当該タスク実行待ちの状態コードに対して 300 足した 400 番台の状態コードで表現される．この状態コードの設計により，タイムアウトで異常終了した取引インスタンスは，300 引いた状態コードで示されるタスクがタイムアウトしたものによるものであることが確認可能である．特定の Task Validator が取引を異常終了させる場合，状態コードに 500 以上を割り当てる．500 以上で示される異常終了の責任は，取引プロセスの定義毎に販売者あるいは購入者どちらの責任であるか定義するものとする．抽象コントラクト中に内部関数として `getState` 関数を定義し，取引インスタンス毎にその状態コードを確認可能にする．

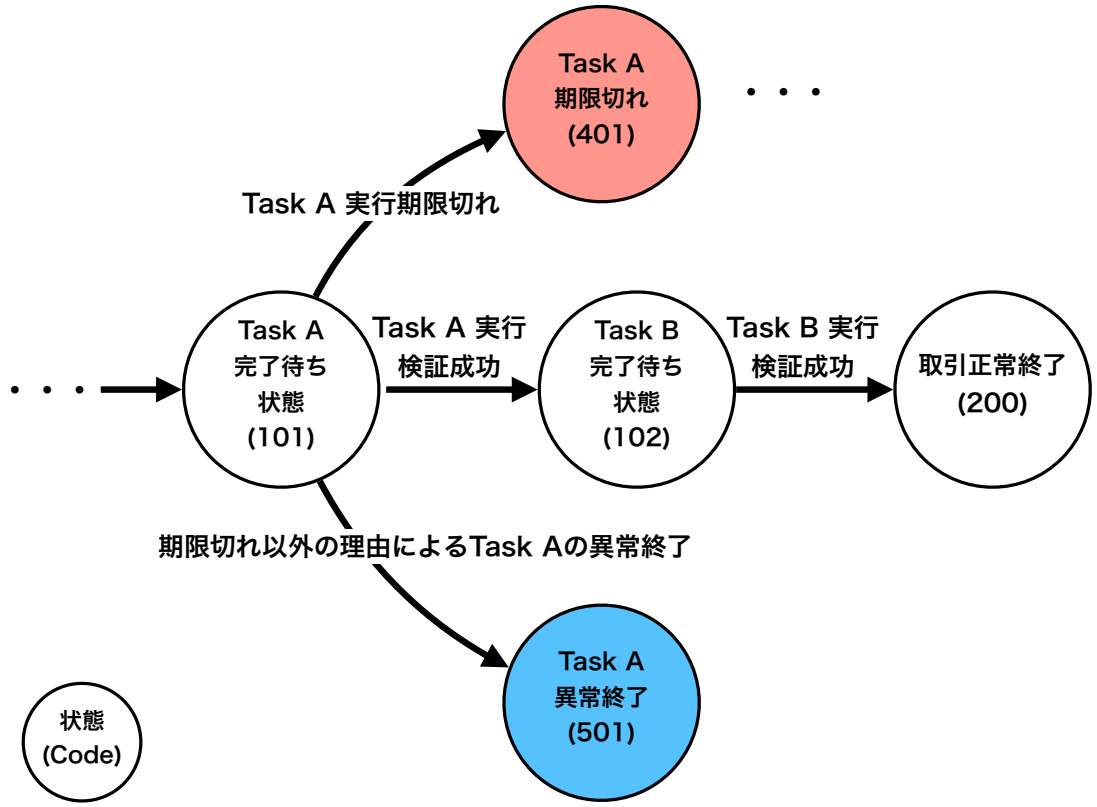

図 6.6　実装における状態遷移: 取引プロセス中の状態コードは 100 番台である．正常終了を示す状態コードは 200 である．特定のタスクがタイムアウトした場合，取引プロセス中の当該タスク実行待ちの状態コードに対して 300 足した 400 番台の状態コードとする．個別の取引プロセスで定義される異常終了は，500 以上の状態コードで示される．

6.4.4　フレームワークを用いた特定取引プロセスの実装

本節では，本実装を用いて特定の Fair Exchange プロトコルを含む取引プロセスを実装することで，フレームワークとして期待通り動作することを示す．既存の Fair Exchange プロトコルである FairSwap の既存の実装をベースに，本フレームワーク上に組み込んだ取引プロセスを設計する [69][*1]．

FairSwap の概要

FairSwap は，動画などのサイズの大きいデータを送信するための Fair Exchange プロトコルである [69]．データの送信を確認するために，FairSwap では特定の要素が含まれるかどうかを効率的に確認できるデータ構造である Merkle 木を活用する．販売者は，送信するデータを分割し，n 個の小さなデータの集合 $E = E_1, E_2 ... E_n$ としたうえで，

[*1] https://github.com/lEthDev/FairSwap

Merkle 木として構成する．購入者は，取引開始前にこの Merkle ルート h を知っているものとする．販売者は，E と，E の Merkle 木を構成するハッシュ値を直列に並べた配列を作成する．次に，販売者は共通鍵暗号によって，鍵 k を用いて当該配列のすべての要素を暗号化する．k を用いて暗号化された要素の配列を Z とする．販売者は，Z を購入者にブロックチェーン外の out-of-bound 通信によって送信する．購入者は Z の各要素より Merkle 木を構成し，そのルートを r とする．次に，購入者は r と h をスマートコントラクトへ送信し，記録する．この時，購入者は代金をスマートコントラクトへと預け入れる．販売者は，記録された r が，自身が送信した Z の Merkle 木のルートであることを確認する．確認できれば，販売者は k をスマートコントラクトへ送信し，記録する．購入者は k をスマートコントラクトから取得し，Z の各要素を復号する．次に，購入者は複合された Z の要素から E の Merkle 木を構成し，ルートが h と一致することを確認することで，期待したデータを受信できたことを確認できる．ルートが h と一致しなかった場合，購入者は E の部分木を含む Z の一部である π をスマートコントラクトへ送信し，データの不一致を主張する．スマートコントラクトは k と r を記録していることから，スマートコントラクトは Z の一部を複合し，不一致を確認できる．スマートコントラクトが不一致を確認した際は，スマートコントラクトは代金を購入者へ払い戻す．π が事前に決められた期限までに購入者から提示されなかった場合，スマートコントラクトは販売者へ代金を払い出す．

フレームワーク上での取引プロセスとしての FairSwap の統合

FairSwap は，以下のタスクから構成される．

- 販売者から購入者への Z の送信
- 購入者からスマートコントラクトへの r，h および代金の送信
- 販売者からスマートコントラクトへの k の送信
- 必要に応じて，購入者からスマートコントラクトへの π の送信

図 6.7 に，これらのタスクを含む取引プロセスを示す．商品が販売中であることを示すために，販売者は h を事前に公開しているものとする．当該商品を購入する購入者は，取引を開始するために h をスマートコントラクトへ送信し，代金を預け入れることで購入のオファーを示す．販売者はオファーを確認し，承認する場合，承認をスマートコントラクトへ記録する．販売者は商品を Z へエンコードし，out-of-bound 通信によって購入者へ送信する．次に，購入者は r を Z から計算し，r を FairSwap の開始を示すためにスマートコントラクトへ送信する．販売者は r が Z の Merkle 木のルートであることを確認する．確認できなかった場合，販売者は FairSwap の開始を拒否し，取引を異常終了させる．確認できた場合，販売者は k をスマートコントラクトへ送信し，購入者が取得可能

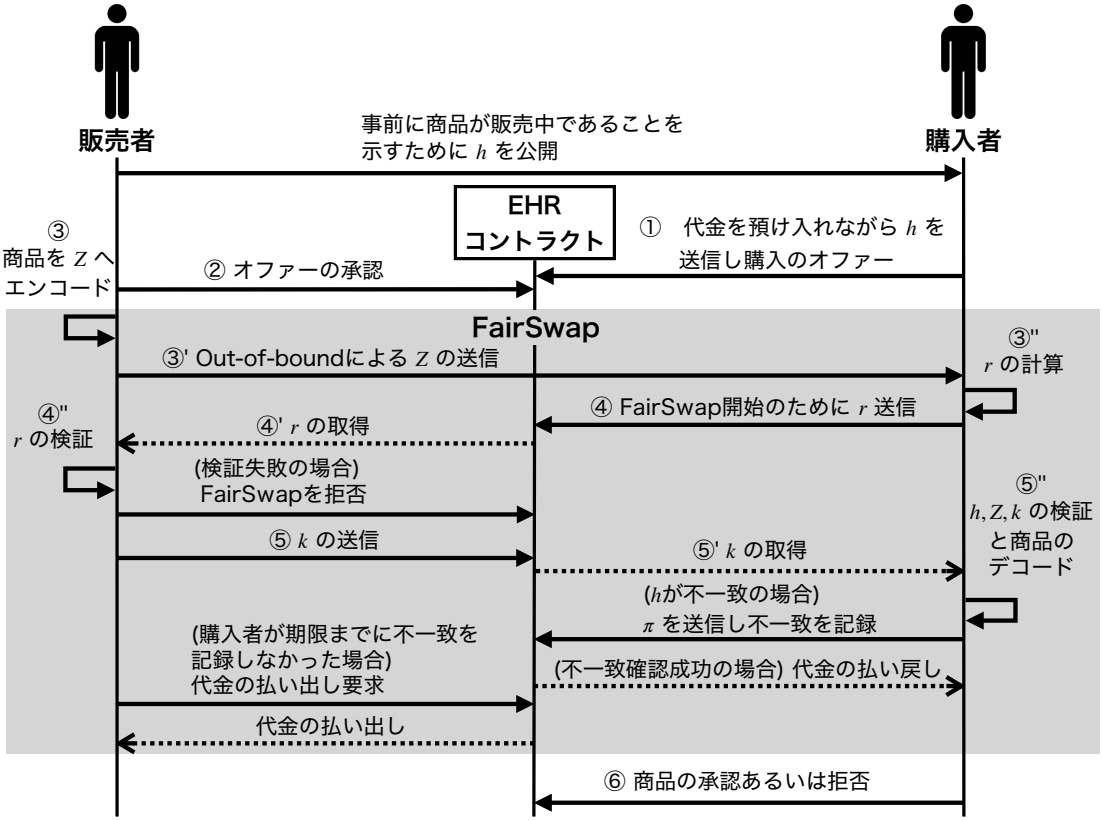

図 6.7　FairSwap を用いた取引プロセス: EHR コントラクト上で FairSwap の各ステップを実行し，商品の送受信を確認する．

にする．購入者は k を取得し，h，Z，および k から商品をデコードする．E の Merkle 木のルートが h と不一致であれば，購入者は π をスマートコントラクトへ送信する．スマートコントラクトは π により不一致を確認した場合，スマートコントラクトは代金を購入者へ払い戻し，取引を異常終了させる．π の確認がタイムアウトした場合，販売者は代金の払い出しをスマートコントラクトへ要求できる．FairSwap のフローに加え，購入者は自身の基準に併せて商品を承認あるいは拒否を記録する．購入者が商品を承認した場合，スマートコントラクトは取引を正常終了させる．

本プロセスにおいてオファーが承認される前にタイムアウトした場合は，取引条件に合意していないものとみなせる．したがって，終了状態の責任はどちらの責任でもないと解釈できる．購入者による r の登録がタイムアウトした場合，販売者が Z を送信しなかったケースと，販売者は Z を送信したが購入者が r を送信しなかったケースが考えられる．したがって，当該ケースでは終了状態だけではどちらの責任であるかは判断できない．そのため，もし販売者がオファーを承認したが購入者が r を送信せずタイムアウトした場合，取引は両者の責任によって異常終了したと定義できる．商品が拒否された場合，商品は販売者によって購入者に送信されたが，商品の質やその他の理由によって紛

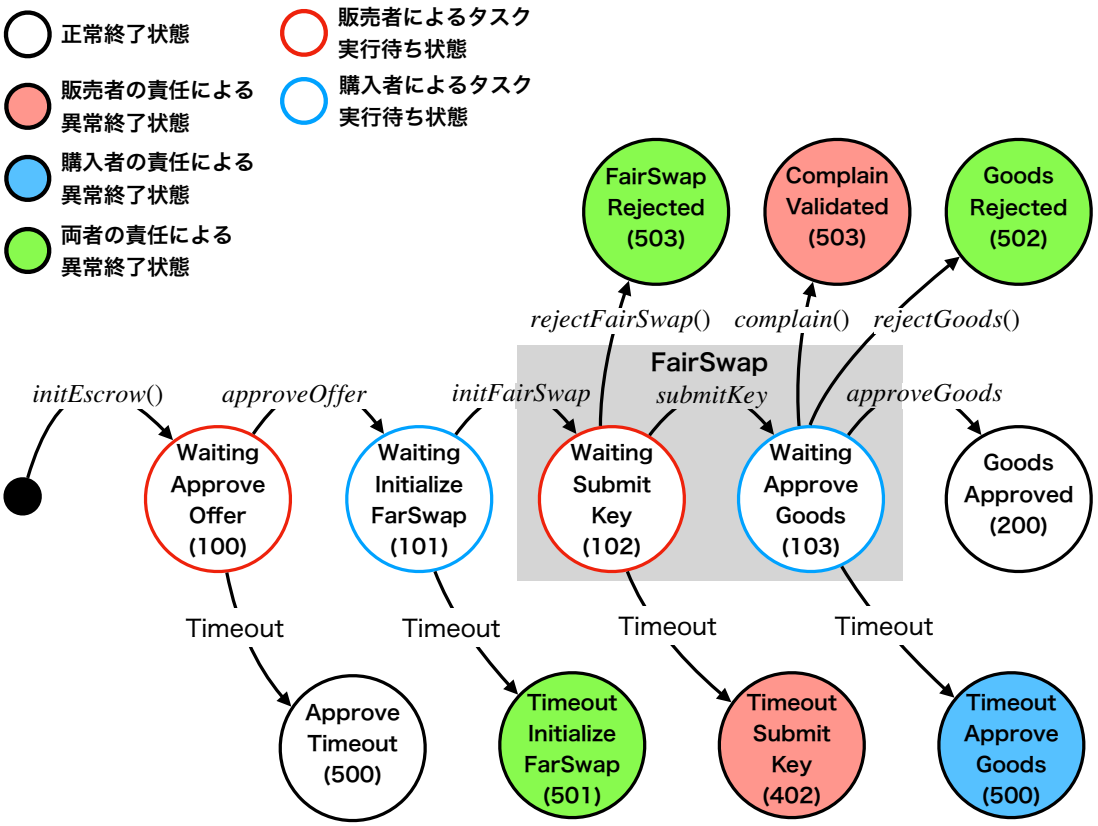

図 6.8　FairSwap を用いた取引の状態遷移

争状態に陥ったケースが考えられる．6.1 節で議論したように，紛争状態はスマートコントラクト上の状態のみではどちらの責任であるかは判断できない．したがって，当該状態では取引は両者の責任にによって異常終了したと定義できる．

提案フレームワーク上での実装

前節で議論した FairSwap を用いた取引プロセスを，本フレームワークを用いて実装した．図 6.8 に実装した EHR コントラクトにおける取引の状態遷移を示す．

購入者が取引を開始する際，購入者は h を `initEscrow` 関数の引数として入力する．EHR コントラクトは，取引インスタンスを作成し，h を当該インスタンスに関連づけた形で保存する．前節で議論した取引プロセスを実行するために，以下の関数を Task Validator として実装した．

- **approveOffer function**: Waiting Approve Offer State において，販売者が販売者からのオファー承認を示すための関数．
- **initFairSwap function**: Waiting Initialize FairSwap State において，購入者が r を EHR コントラクトへ記録するための関数．

- **submitKey function**: Waiting Submit Key State において，販売者が k を EHR コントラクトへ記録するための関数.
- **approveGoods function**: Waiting Approve Goods State において，購入者が商品を承認するための関数.

また，取引を異常終了させるために以下の関数を定義した.

- **rejectFairSwap function**: Waiting Submit Key State において，販売者が FairSwap の拒否を示すための関数.
- **complain function**: Waiting Approve Goods State において，購入者が販売者が誠実に FairSwap を実行していないことを主張するための関数．本関数は，π を入力として受け付け，すでに EHR コントラクト上に記録されている h，r および k を用いて，送信されたデータの不一致を確認する.
- **disburse function**: 販売者が報酬の払い出しを要求する関数．販売者は Waiting Approve Goods State で販売者が complain 関数を実行できる期限以降に本関数を実行できる.

以上の実装を通じて，本フレームワーク上で FairSwap を含む取引プロセスを実装可能なことを確認した．本実装では，FairSwap は提案時のオリジナル実装を元に構成した．FairSwap に対して送信されるデータの機密性を含めた改善の提案がある [90, 91, 92]. これらの提案は，本実装における `complain` 関数を改変することで実装可能である．例えば，FairSwap 自体に脆弱性が発見されたなどの理由によって改変が必要になった場合，取引プロセス中のタスクを示す Task Validator を改変することで，プロトコルを改変できる．本実装の今後の展望として，終了状態の責任が両者の責任とならないように FairSwap を改善することなどが考えられる.

6.4.5 パフォーマンス分析

本節では，パフォーマンス分析として，本フレームワーク上で取引を実行する際の手数料を計測し，本実装が Ethereum 上で実行可能なことを分析する．Ethereum では，スマートコントラクトのデプロイ，あるいはデータをスマートコントラクトへ記録する際に仮想通貨立てで手数料を支払う必要がある．手数料の金額は，EVM 上でのプログラム実行ステップ数によって決定され，EVM 上の各 opcode に対しての金額が “Gas” と呼ばれる単位でハードコードされている．したがって，本フレームワークでは EHR コントラクトのデプロイ時および `initEscrow` 関数，`processTask` 関数を実行する際に手数料が発生する．スマートコントラクト上のデータの読み出しには手数料を必要としないため，`getHistoryAsSeller/getHistoryAsBuyer` 関数の実行には手数料を支払う必要は

ない．また，

フレームワークのプロトタイプ実装では，取引プロセスに含まれるタスクの数と，`processTask` 関数で呼び出される Task Validator は，`taskSequence` によって決定される．`taskSequence` は，EHR コントラクトのコンストラクタ内で定義される．また，`initEscrow` 関数と `processTask` 関数は，コンストラクタ内で定義される *Transaction-Definition* を用いる．したがって，本フレームワークの手数料は，EHR コントラクトで定義される取引プロセスの以下の要素によって決定される．

- 取引プロセス中に含まれるタスクの数
- Task Validator の処理内容

フレームワークとしてのスケーラビリティを評価するために，単一の取引プロセス中に含まれるタスクの数に応じた手数料の変化を分析した．EHR コントラクトを Ethereum におけるスマートコントラクト開発環境である HardHat 上で実行し，手数料を計測した [93]．タスクの数による変化のみを計測するために，特定の Task Validator を実装し，当該タスクを繰り返し実行する取引プロセスを定義した．したがって，Task Validator を 1 回実行するための手数料は全ケースにおいて一定である．

実験として，EHR コントラクトのデプロイ時および `initEscrow` 関数，`processTask` 関数実行時の手数料を取引プロセスに含まれるタスクの数を 0 タスク (ベースライン)，1 タスク，5 タスク，10 タスクのケースにおいて計測した．図 6.9 に結果を示す．実験の結果，取引プロセス中に含まれるタスクの数に応じて，各関数の実行手数料が増加することが明らかとなった．また，Ethereum 上でスマートコントラクトを実行する際の手数料の上限は $30 \times 1e6$ Gas であることから，10 タスクのケースにおいても本フレームワークは実行可能である．`processTask` 関数においては，手数料はその処理内容によって変化するため，EHR コントラクト内でより複雑な暗号プロトコルを用いた場合，より多くの手数料が必要となるだろう．一方，スマートコントラクトの実行手数料を削減する試みは複数提案されており，本フレームワークにおいても適用可能である [94]．フレームワークの実装の手数料に関する最適化や，手数料を Ethereum よりも必要としないブロックチェーン上での実装は，今後の展望の 1 つである．

6.5 脅威分析と要件の充足

本節では，本フレームワークに対する脅威分析と，要件の充足について議論する．まず，本フレームワークの限界を示すために想定される脅威に対する対応を議論する．次に，本フレームワークが要件を満たすことを分析する．

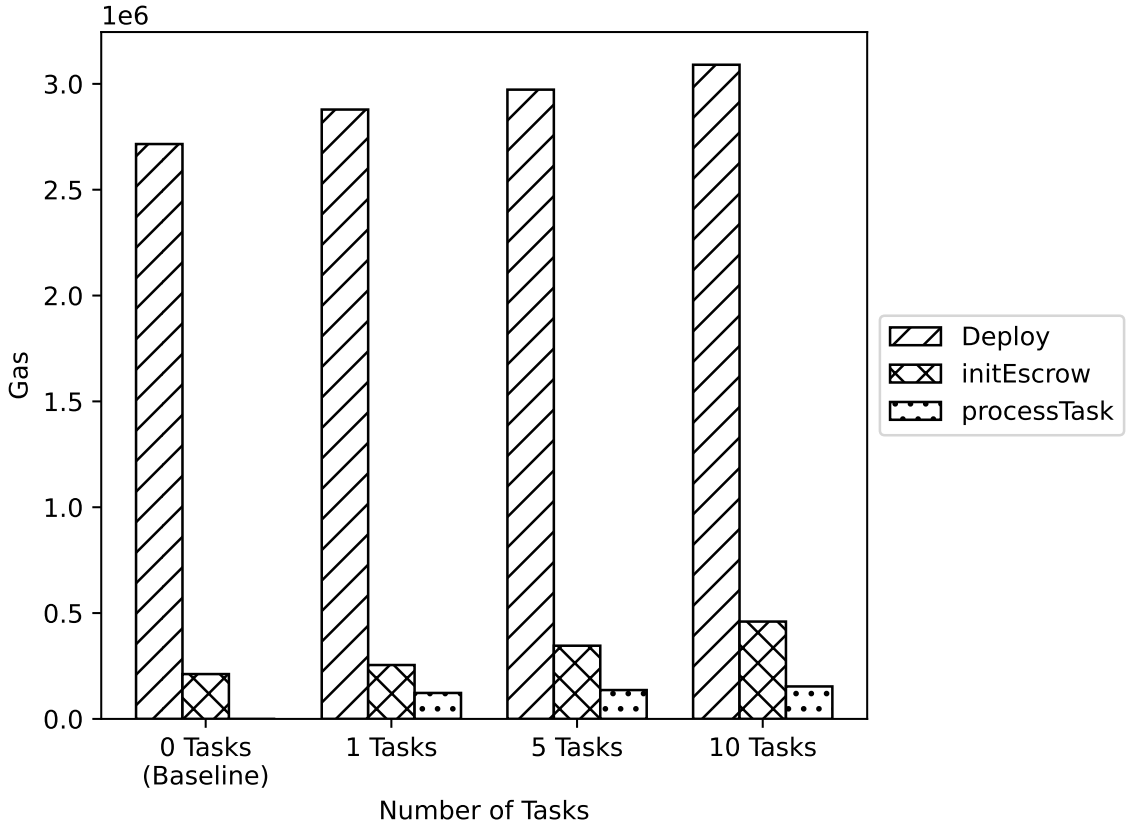

図 6.9　取引フレームワークの手数料計測結果: 本フレームワークを用いた EHR コントラクトでは，より多くのタスクが取引プロセス中に含まれればより多くの手数料が必要となることが明らかとなった.

6.5.1　脅威分析

本節では，本フレームワークの有効性と限界を示すために想定される脅威を議論する．本フレームワークは特定のエンティティが関与した過去の取引の結果を確認可能にし，紛争に陥る傾向を評価することを目的とする．したがって，攻撃者が作為的に過去の取引の終了状態の傾向を操作する脅威を議論する．

架空の取引履歴の作成

攻撃者が実施していない取引の履歴を作成し，終了状態の傾向の操作を試みることが考えられる．本フレームワークに沿った EHR コントラクト上で取引履歴を作成するには，販売者と購入者は EHR コントラクト上で定義された取引プロセスを実施する必要がある．ブロックチェーンの改ざん困難性より，攻撃者は履歴を恣意的に書き換えることはできない．また，EHR コントラクトは定義されたプロセスにしたがって実施された取引の履歴のみを記録するため，本脅威は本フレームワークで実装された EHR コントラク

トに対しては実行不可能である．

取引履歴の選択的開示

攻撃者が過去の取引を異常終了した履歴を持つとき，攻撃者が異常終了の履歴を提示しないことが考えられる．本フレームワークに沿った EHR コントラクトでは，将来の取引相手は EHR コントラクトに対して，傾向を確認したいエンティティの識別子である Ethereum Address を用いて過去の取引履歴を取得する．EHR コントラクトは，指定された識別子に紐づくコントラクト内で保存された全ての履歴を出力するように設計されている．したがって，攻撃者は特定の EHR コントラクト内に保存された履歴の中から部分的に開示はできない．さらに，ブロックチェーンの特性により，全ての取引履歴は任意のエンティティに開示されており改ざん困難である．そのため，履歴を選択的に開示することは困難である．

共謀による履歴の作成

販売者と購入者が共謀し，正常終了する取引を実施し，それぞれの取引終了状態の傾向を操作することが考えられる．一方，履歴を作成するためには，EHR コントラクト上で定義された取引プロセスを実行する必要がある．これは，取引が正常終了したことを示しており，誠実なエンティティが取引を正常終了させたことと履歴の上の差異はないことを意味する．したがって，過去の関与した取引結果の検証可能性を確保する本ケーススタディの目的に対しては，本脅威は脅威として成立しない．それぞれの結果を集約し，エンティティの傾向をどのように評価するかは，本ケーススタディの範囲外である．今後の検討課題として，特定のエンティティを評価を計算するレピュテーションシステムの議論などと組み合わせることが考えられる．例えば，特定の 2 者間で繰り返し取引が実施された履歴に対しては，共謀している可能性があるとみなし，評価する際に考慮することが考えられる．

シビル攻撃

攻撃者が複数の Ethereum Address を作成し，2 つの Ethereum Address 間で取引を実施することで，正常終了の履歴を恣意的に作成することが考えられる．この脅威は，直接的には本フレームワーク単体では対処できないが，将来の取引相手が履歴を確認する際に，特定の Ethereum Address に紐づくエンティティのアイデンティティを確認することで，一定緩和が期待できる．例えば，特定の Ethereum Address に紐づくエンティティを Know Your Customer (KYC) などの手法を用いて確認することが考えられる．特定の Ethereum Address に紐づくエンティティのアイデンティティ検証は本ケーススタディの範囲外であるが，アイデンティティ検証との組み合わせは本フレームワークの

重要な検討課題の 1 つである．

履歴の白紙化

攻撃者が過去の取引を異常終了させた履歴を持つとき，攻撃者が新たな Ethereum Address を作成することで過去の履歴を白紙化することが考えられる．一方，履歴を白紙化した場合，正常終了の履歴も同時に失われる．したがって，正常終了の履歴が失われることは，本攻撃のディスインセンティブとなる．具体的にどのようにディスインセンティブになるかは，履歴がどのように評価されるかに依存する．評価の方法をデザインすることで，単一の識別子を用い続けるインセンティブを設計することが検討課題として考えられる．この観点からも，本脅威を緩和するためにアイデンティティ検証と履歴の評価手法を検討することは，重要な検討課題の 1 つである．

6.5.2 要件の充足

本節では，6.3.1 節で示した要件を，本フレームワークが充足したかどうかを議論する．

エスクロー実施に対する要件

ER-1 は，代金を購入者が取引条件を提示すると共に EHR コントラクトへ預け入れるため，満たされている．**ER-2** は，取引プロセスの定義に沿って，特定のタスクが完了した時のみ販売者へ代金を払い出すため，満たされている．**ER-2'** は，取引が異常終了した際は，購入者は代金を払い戻せるため，満たされている．

取引履歴の閲覧に対する要件

本ケーススタディでは，Ethereum Address をエンティティの識別子として，販売者あるいは購入者として関与した取引の履歴を取得可能な EHR コントラクトをデザインした．Ethereum Address は，検証鍵のハッシュ値であるため，特定の Ethereum Address から取得可能な履歴を作成するためには，当該検証鍵に対応する署名鍵を用いた署名付きデータをブロックチェーン上へ記録する必要がある．したがって，**HR-1** は達成されている．次に，EHR コントラクトに対するそれぞれの操作は，同様に署名付きデータを用いて行われるため，EHR コントラクトに記録される取引インスタンスの状態はブロックチェーンの特性により改ざん困難である．したがって，**HR-2** は達成されている．また，将来の取引相手が取引履歴と取引インスタンスの状態を確認可能に設計した．状態は EHR コントラクトで定義されたステップでのみ遷移するため，**HR-3** は達成されている．最後に，状態は取引プロセスのいずれのタスクの状態から異常終了したかを確認可能に設計した．また，各異常終了の状態における異常終了の責任は，取引プロセスの定

義に併せて明確化される．したがって，**HR-3'** は達成されている．

フレームワークとしての要件

本フレームワークは，取引プロセスに含まれるタスクとその完了の確認を定義できる．したがって，**FR-1** は満たされている．**FR-2** は，取引プロセス中の状態を特定のタスクの実行待ち状態と定義し，タスクの順序を定義可能にすることで満たされている．*Task execution interface* は，特定のタスクの完了が Task Validator によって確認された時のみ，取引インスタンスの状態を次のタスクの待ち状態に遷移させる．したがって，取引インスタンスの状態はスマートコントラクトのコードに定義された形式でのみ遷移するため，**FR-3** は満たされている．

以上の分析により，本フレームワークは全ての要件を満たしたと結論づける．

6.6 本ケーススタディの今後の展望

本節では，本ケーススタディの今後の展望を述べる．

6.6.1 フレームワークの標準化

本ケーススタディでは，過去の関与した取引の結果をクレームとして，クレーム検証を実現するために取引プロセスの履歴を確認可能な取引フレームワークを設計した．Ethereum では，特定のアプリケーションを実装する複数のコントラクトの相互運用性を確保するために，共通のインターフェースの定義などの標準規格が定義されている [95]．たとえば，Non-fungible Token (NFT) と呼ばれるタイプのトークンを実装するための標準インターフェースを定義する標準規格がある [96, 97]．このような標準と同様に，本フレームワークも標準として定義することで，EHR コントラクトを通じて様々な取引プロセスの結果を，共通のインターフェースから検証可能にするエコシステムが実現可能と考えられる．

6.6.2 取引プロセス定義の表現

本ケーススタディでは，取引プロセスは分岐のない直線的なプロセスを想定した．実際の様々な取引のプロセスを考慮すると，分岐のある取引プロセスも考えられるだろう．たとえば，特定の状態におけるタスクの実行結果や取引インスタンス中の特定の値によって，複数の状態へ遷移しうるプロセスが考えられる．より，柔軟に様々な取引プロセスに対応するには，Task Validator の結果に応じて状態遷移した際の取引の結果の責任の整理を議論する必要があるだろう．

本ケーススタディの関連の議論として，特定のビジネスプロセスの監査可能性を確保するために，ビジネスプロセスを機械可読な形で記述する手法である BPMN (Business Process Model and Notation) を用いて，ブロックチェーン上でビジネスプロセスを実行する議論がある [98, 99, 100]．これらで議論される BPMN で記述されたビジネスプロセスをスマートコントラクトへ変換する手法を参照しながら，様々な取引プロセスを表現可能にすることが考えられる．

また，本ケーススタディにおける実装では，Ethereum 上で Solidity を用いて実装した．そのため，Task Validator によって確認できるタスクの完了の確認は Solidity や Ethereum の制約を受ける．たとえば，Task Validator 中で複雑な計算によってタスクの完了を確認する場合，より多くの手数料を支払ってタスクを実行する必要がある．一方，本フレームワークはスマートコントラクトが実装可能な任意のブロックチェーン上で実現可能であるため，より手数料の側面で効率の良いブロックチェーン上で実装することが考えられる．

6.6.3 レピュテーションシステムへの統合

Seller and Buyer's Dilemma の緩和のためには，取引の結果を検証したのち，各結果の示す責任とそれに基づいて対象エンティティを評価する必要がある．特定のエンティティを評価するために，様々な情報を統合してスコアを算出するなどのレピュテーションシステムの議論がある [101]．本フレームワークが提供する履歴を元に，対象エンティティを評価するためにはこれらのレピュテーションシステムと統合することで適切な評価手法を検討する必要があるだろう．

6.6.4 エンティティと識別子の紐付け

本フレームワークの実装では．エンティティの識別子として Ethereum Address を仮定し，取引の履歴は Ethereum Address をキーとして取得可能にデザインした．しかし，6.5.1 節で議論したように，いくつかの脅威は識別子に紐づくエンティティのアイデンティティを検証しなければ対処できない．したがって，Know Your Customer (KYC) の手法などを用いて，識別子に紐づくエンティティのアイデンティティを検証することで，識別子とエンティティの紐付けを確認方法を検討する必要があるだろう．

6.7 本章のまとめ

本章では，Shinken モデルのケーススタディとして，商取引の結果をクレームとして検証するための検証基準の構成と，検証基準に基づき結果を検証可能な取引フレームワー

クを議論した．商取引の結果は，商取引のプロセス定義に沿って，当該プロセスに含まれるタスクを実行した結果であることから，取引プロセスの履歴を確認可能にすることで，検証基準を構成できると整理した．その上で，ブロックチェーンを用いて定義されたプロセスに沿った履歴が改ざん困難であることを仮定し，多様な取引プロセスを定義可能な取引フレームワークを設計した．本フレームワークを用いて，取引プロセスを定義，実行することで，取引の結果を検証できると結論づけた．本ケーススタディを通じて，Shinken モデルに基づき，取引の結果を検証する基準を構成できたことから，具体的な事例における Shinken モデルの適用可能性を実証した．

次章では，Shinken モデルの 2 つ目のケーススタディとして，デジタル証明書を用いたクレームの検証基準の更改を取り上げる．

第 7 章

ケーススタディ 2: デジタル証明書を用いたクレーム検証基準の更改

本章では，Shinken モデルに基づいて検証基準を更改するケーススタディとして，協調プロセスの結果得られる実績を対象に，デジタル証明書を用いたクレーム検証を取り上げる．

7.1 背景: 協調プロセスの結果である実績の証明書

本章では，主張者が何らかの協調プロセスの結果獲得する実績を主張するために，証明書を活用するケースに着目する．協調プロセスの結果得られる実績の例として，大学における単位取得が挙げられる．講義を協調プロセスと捉えると，講師は学生に対して課題を出題し，学生は対応するレポートを提出する，などのタスクを含むプロセスであると整理できる．この課題出題と提出の一連のプロセスを通じて，学生が当該講義の単位取得に足ると評価されれば，講師は当該単位の取得を記録する．大学はこの記録を確認することで，当該学生の単位取得を検証し，当該学生に対して証明書を発行できる．検証者は，当該証明書の提示を受け，当該単位の取得を検証する．

2.3.3 節で議論した証明書モデルに基づき，協調プロセスの結果獲得する実績をクレームとするやり取りの概要を図 7.1 に示す．このやり取りには，以下のエンティティが登場する．

- **参加者**: 協調プロセスに参加するエンティティ．
- **主張者**: 協調プロセスの結果，実績を獲得する参加者の 1 人．実績を示す証明書の発行を受け，実績を持つことを主張する．

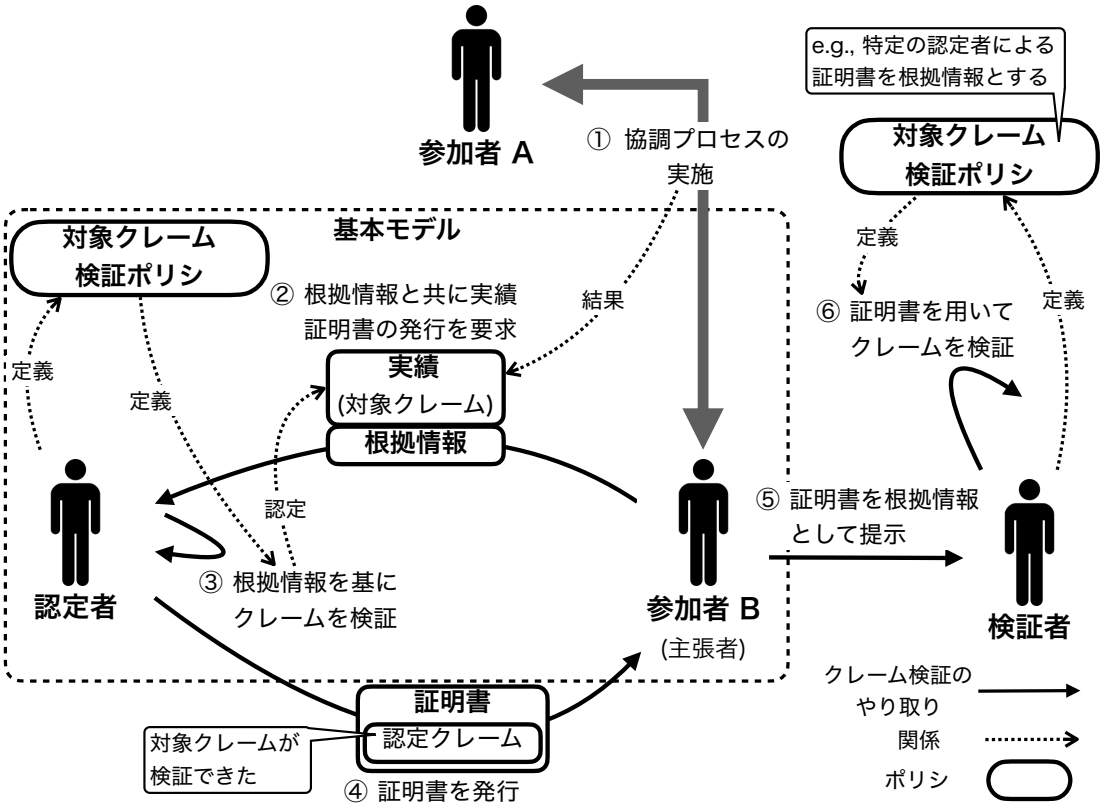

図 7.1　協調プロセスの結果獲得する実績をクレームとするやりとりの概要: 本図は，参加者 B が実績を獲得し，主張者となるケースを示す.

- **認定者**: 協調プロセスの実施を確認し，実績を検証することで，証明書を発行するエンティティ.
- **検証者**: 実績を示す証明書を根拠情報として受け入れ，実績を検証するエンティティ.

まず，参加者は協調プロセスを実施し，その結果主張者は実績を獲得する．検証者は，特定の認定者による証明書を当該実績の根拠情報とすることを対象クレーム検証ポリシに定める．したがって，認定者に対して証明書の発行を要求するため，基本モデルに基づき主張者は実績を主張し，認定者は実績を検証する．認定者は，当該協調プロセスとその結果を，自身が実績と認める基準に達していることを検証する．認定者による実績の検証基準は，実績を対象クレームとする対象クレーム検証ポリシに定められる．検証に成功した場合，認定者は証明書を発行する.

7.2 検証基準の構成と更改の方向性

5.4.3 節で議論したように，デジタル証明書を用いた場合，“認定者が証明書を発行したならば，認定者が証明対象のクレームを検証した” という命題が真であることを仮定することで，検証者は検証基準を構成できる．協調プロセスの結果である実績をクレームとする時，“認定者が協調プロセスをそのポリシに併せて実績に足ると検証した上で，証明書を発行していること” を検証者は信頼する．これは，5.4.3 節において議論した署名の意図が，“認定者がその検証ポリシに沿って，協調プロセスが実績に足ると検証した” という命題であることを示す．

しかし，当該命題に対する仮定が破られる可能性を考慮すると，検証者が証明書に含まれるデジタル署名の検証に成功したとしても，直接的には “認定者が対象クレームを検証した” ことは演繹できない．たとえば，悪意のある認定者によって，対象クレームを検証しないままに証明書が発行されるケースが考えられる．具体的には，ディプロマミルと呼ばれる，正規の学位授与のプロセスを経ていない学生に対して，証明書を発行する機関の存在が知られている [102, 103, 104]．また悪意によらずとも，認定者の証明書発行システムが攻撃を受ける，あるいはヒューマンエラーによって，検証されていないクレームに対する証明書を発行するケースが考えられる．たとえば，単位取得の証明書のケースにおいて，講師が誤って単位取得を記録する可能性がある．これらのケースでは，検証者は証明書のデジタル署名が検証でき，認定者が発行した証明書であると確認できたとしても，証明書で示されるクレームの妥当性を正しく判断できず，偽陽性あるいは偽陰性となるケースが考えられる．

ここで，検証者視点で，“認定者が特定の根拠情報を基に実績を検証したこと” が確認できれば，先述の危殆化した証明書発行プロセスを検知できると考えられる．そのため，先述のケースに対応するには，証明書に示される実績に対する検証基準の中で，“認定者がそのポリシに沿って証明書発行前に当該実績を検証したこと” を確認することが好ましい．したがって，検証基準を更改し，“認定者が証明書の発行にあたり，特定の根拠情報を確認したこと” の検証を追加する．図 7.2 に更改前後の検証基準を示す．また，付録 A.5 に，本節の議論に基づいた更改後の検証基準の Prolog による実装を示す．

7.3 協調プロセス確認の検証手法の設計

図 7.1 で示したやり取りでは，認定者は実績に至る協調プロセスを確認した上で証明書を発行する．したがって，本ケーススタディでは，当該協調プロセスを認定者が確認したことを，検証者視点で検証可能にする手法を検討する．なお，実績認定に至るかどうかの評価は，具体的な実績が何であるかに依存するため，ここでは “協調プロセスを認定者が実績に至ると評価したこと” 自体の妥当性判断は検討の範囲外とする．

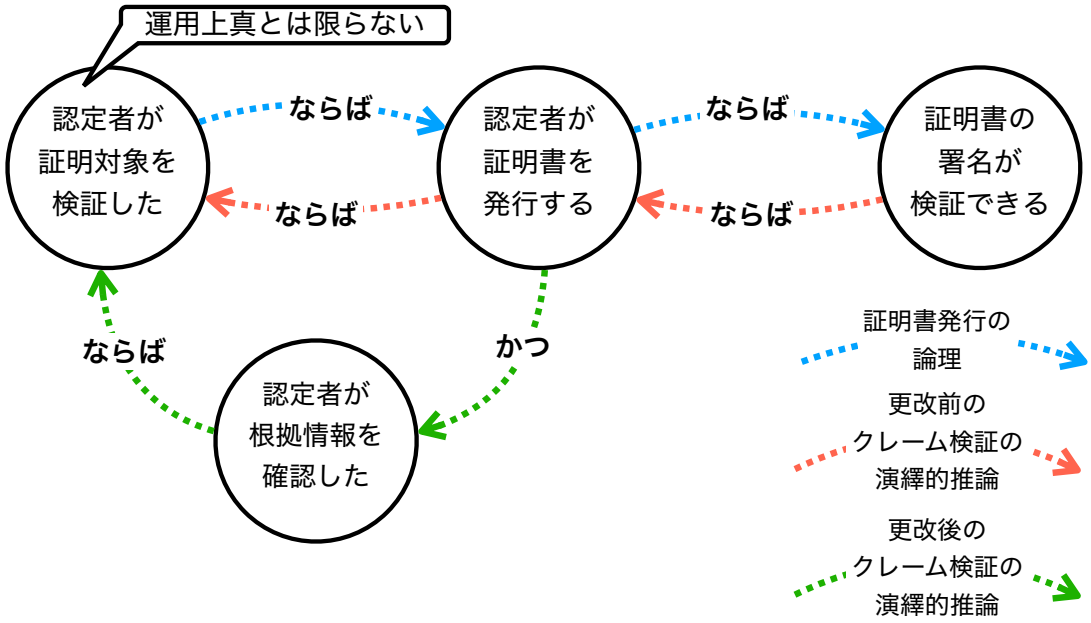

図 7.2　根拠情報の確認を追加する検証基準の更改: 証明書の発行前に “認定者が根拠情報を確認したこと” を検証基準に追加することで，危殆化した証明書発行プロセスを検知可能にする．

本節では，まず，協調プロセスを定義するための要素技術としてビジネスプロセスを機械可読な形式で記述する Business Process Model and Notation (BPMN) について概説する．次に，“実績に至る協調プロセスの履歴を，当該実績の根拠情報として認定者が確認した上で証明書を発行したこと” を検証者が検証可能な手法を提案する．本手法では，協調プロセスを機械可読な形式で定義し，定義に沿ったプロセス中のやり取りの履歴を記録する．その上で，主張者は認定者に当該履歴を根拠情報として提示し，認定者が当該履歴に紐づく形で証明書を発行することで，“特定の協調プロセスの履歴を根拠情報として認定者が確認したこと” を検証者が検証可能にする．

7.3.1　要素技術: Business Process Model and Notation (BPMN)

本節では，本ケーススタディにおいて設計する手法の要素技術となる Business Process Model and Notation (BPMN) について概説する [98]．BPMN は，機械可読な形式でビジネスプロセスを定義するためのモデリング手法である．BPMN の標準では，BPMN で記述されたプロセス定義を XML 形式へシリアライズするために XML Schema を定義している．

BPMN を用いて，複数エンティティによる協調プロセスを記述できる．図 7.3 に協調プロセスの記述例を示す．“レーン (Lane)” はエンティティを示し，レーン上に記述され

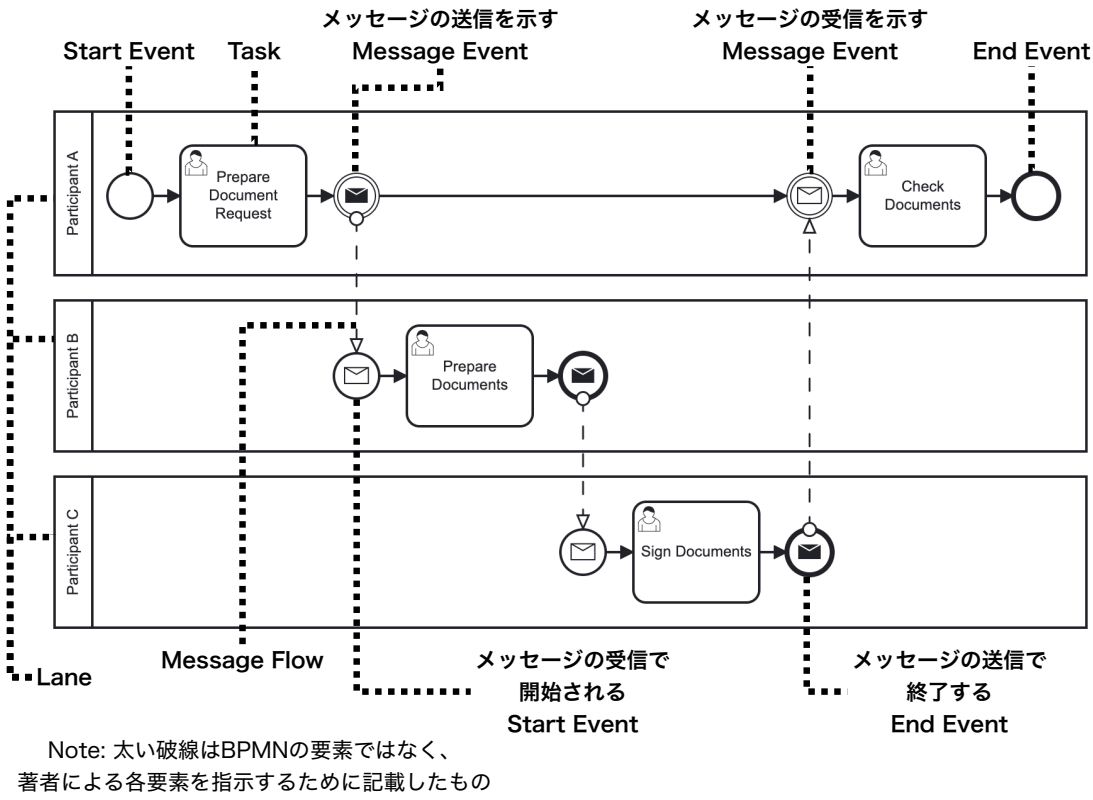

図 7.3　Business Process Model and Notation (BPMN) で記述された協調プロセスと要素

た “タスク (Task)” は当該レーンが示すエンティティに割り当てられたものであることを示す．エンティティ同士の通信は各レーン上の “イベント (event)” を “メッセージフロー (message flow)” で繋ぐことによって表現できる．BPMN における各要素はそれぞれ識別子を持ち，単一のプロセス定義の中で一意に識別できる．

7.3.2　要件定義

“認定者が特定の協調プロセスの履歴を根拠情報として確認した上で証明書を発行したこと” を検証可能にするため，検証者視点で以下が確認できることを要件として定義する．

- **R-1**: 主張者を含む複数の参加者が送信するメッセージと，その順序を含む協調プロセスの定義が BPMN などで記述され，検証者が確認可能であること．
- **R-2**: 協調プロセスの定義に含まれるメッセージが，いずれの参加者が送信するべきものとする割り当てを，検証者が確認可能であること．
- **R-3**: メッセージの送信を割り当てられた参加者が，当該メッセージを送信していることを，検証者が確認可能であること．

- **R-4**: 認定者が特定の協調プロセスの履歴を根拠情報として，実績を検証した上で発行した証明書であることを，検証者が確認可能であること．

7.3.3 前提

本手法を設計する上での前提を述べる．まず，主張者を含む協調プロセスの参加者は協調プロセス中の全てのメッセージを “プロセス履歴 (Process History)” として記録し，保管するものとする．次に，協調プロセスの参加者同士の識別子はお互いに確認できているものとする．また，主張者は直接証明書などのやり取りをするエンティティであること，認定者は広く証明書を発行するエンティティであることから，検証者視点では主張者と認定者の識別子は確認できているものとする．主張者，参加者，認定者の各エンティティの識別子は，当該エンティティが作成したデジタル署名を検証できる検証鍵と紐づく．それぞれの検証鍵に対応する署名鍵は，各エンティティが安全に管理することを仮定し，鍵管理の具体的な手法は本ケーススタディの範囲外とする．また，それぞれの識別子に紐づくエンティティの検証鍵以外のアイデンティティの確認方法，および適切な開示範囲は，具体的なユースケースに依存するため本ケーススタディの範囲外とする．また，各エンティティ間の通信は，メッセージの到達が保証される安全なコミュニケーションチャネルを用いるものとする．加えて，各メッセージは 3.1.2 節で定義した署名付きデータであり，検証鍵の参照として送信者の識別子を含む．ここで，各メッセージにおける署名の意図は，当該メッセージを送信したことであるとする．したがって，各メッセージの署名を検証することで，識別子で示される送信者が当該メッセージを送信したことを確認できる．

7.3.4 提案手法

本ケーススタディでは，協調プロセス参加者間で合意されたプロセスの定義に従ったメッセージの履歴を，当該プロセスの結果である実績の根拠情報として扱う．その上，認定者が履歴と紐づけた形で証明書を発行する手法を提案する．本手法では，主張者は同じプロセス履歴を認定者と検証者それぞれに提示することで，“認定者が当該履歴を根拠情報として実績を検証した上で証明書を発行したこと” を検証者視点で確認可能にする．

メッセージの種類

本節では，本手法におけるメッセージの種類を述べる．本手法では，4 種類のメッセージを定義する．

- **Proposal Message**: プロセス中のタスクおよびメッセージを含む協調プロセス

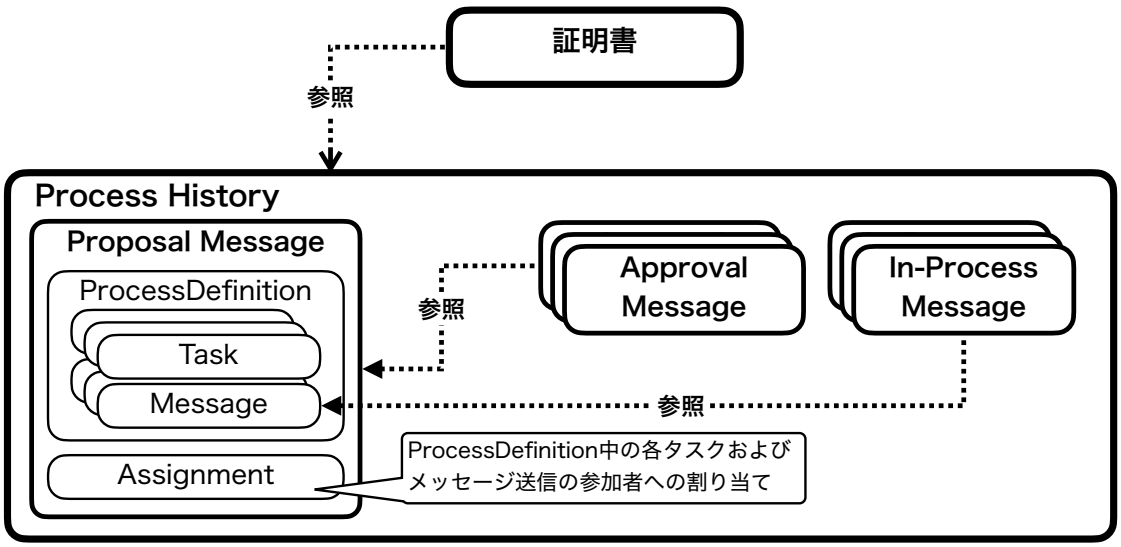

図 7.4　本手法におけるメッセージ同士の関係

の定義と，各参加者へのタスクおよびメッセージの割り当てを示すメッセージ

- **Approval Message**: 参加者が Proposal Message に合意したことを示すメッセージ
- **In-Process Message**: Proposal Message 中の協調プロセスの定義に沿った，プロセス中に交換されるメッセージ

図 7.4 に各メッセージ間の関係を示す． Approval Message は，合意する Proposal Message に対する参照を含む． In-Process Message は，Proposal Message 中のメッセージの定義に対する参照を含む．また，認定者が協調プロセスを通じた実績を検証したことを示す証明書は，各メッセージを全て含むプロセス履歴全体を参照する．

本提案におけるデータフロー

図 7.5 に提案手法におけるデータフローを示す．7.1 節で示したエンティティに加えて，以下の役割を持つ参加者を定義する．

- **提案者**: 協調プロセスの定義を提案する参加者.

本ケーススタディでは，提案手法を提案者，主張者に加えてもう 1 名の参加者による 3 者間での協調プロセスを基に議論する．

まず，提案者は Proposal Message を他の協調プロセス参加者へ送信する．本手法では，各参加者は全てのメッセージを全ての協調プロセス参加者へブロードキャストする．Proposal Message は BPMN などのプロセスモデリング手法で記述されたプロセスの

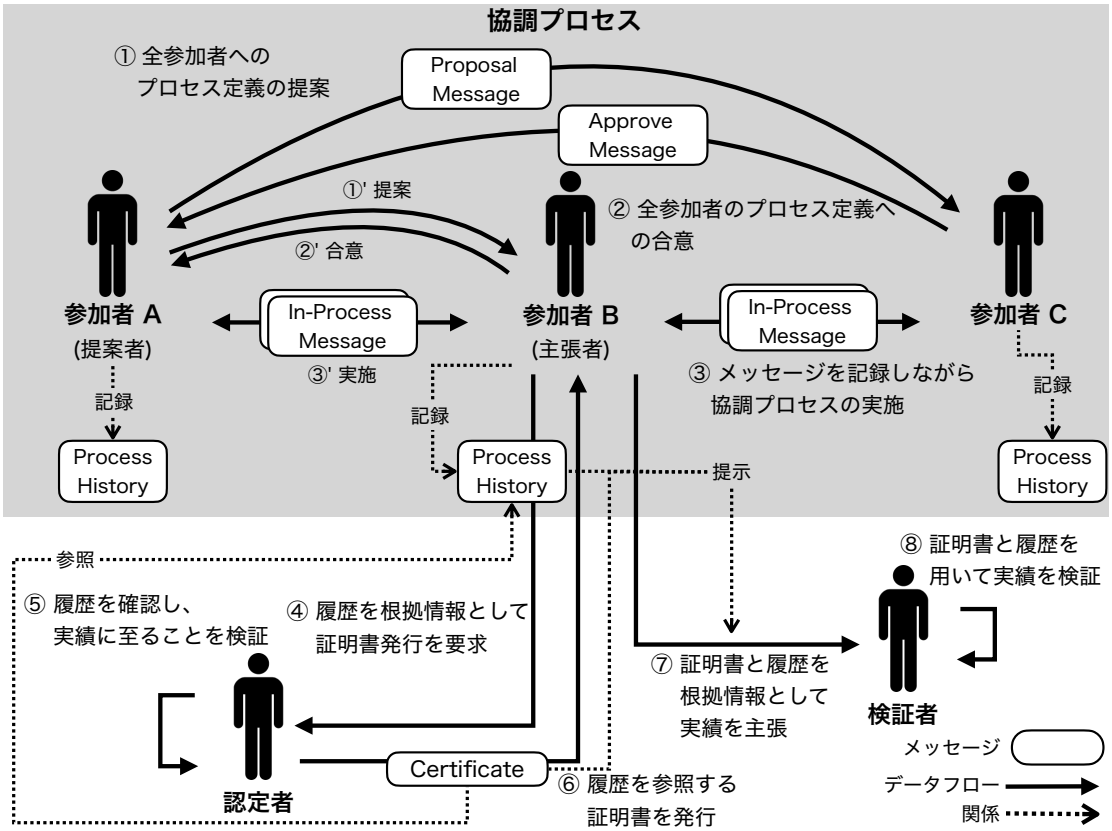

図 7.5　本提案におけるデータフロー: 本図は，3 者間の協調プロセスと，その結果参加者 B が主張者として実績を獲得するケースを示す．各エンティティは協調プロセスの履歴を記録し，保持する．協調プロセスの終了後，参加者 B は認定者に当該プロセス履歴を提示しながら証明書の発行を要求する．参加者 B が実績をクレームとして主張する際は，証明書とプロセス履歴を根拠情報として検証者に提示する．

定義と，定義中の各タスク及びメッセージ送信の参加者への割り当てを含む．Proposal Message の内容に合意する場合，参加者は Approval Message を送信する． Proposal Message と Approval Message の交換によって，協調プロセスの定義全体と各参加者へのメッセージ送信の割り当てが参加者間で合意される．また，Proposal Message への参加者間の合意によって，各メッセージを送信するべき参加者が決定する．Approval Message の交換ののち，参加者間で協調プロセスを実行する．

図 7.6 に，協調プロセス中のデータフローの詳細を示す．前提より，全ての参加者は全てのメッセージを記録し，各メッセージは到達が保証されている．したがって，全ての参加者は当該履歴を用いてプロセスの実施を主張できる．仮に，特定のメッセージの内容をプロセス定義で示される受信者のみが閲覧可能にする場合，送信者は受信者の検証鍵でメッセージのペイロードを暗号化することで秘匿できる．メッセージの内容を秘匿する具体的な方法は本ケーススタディの範囲外とする．

協調プロセスが終了したのち，主張者は認定者に対して実績を示す証明書の発行を要

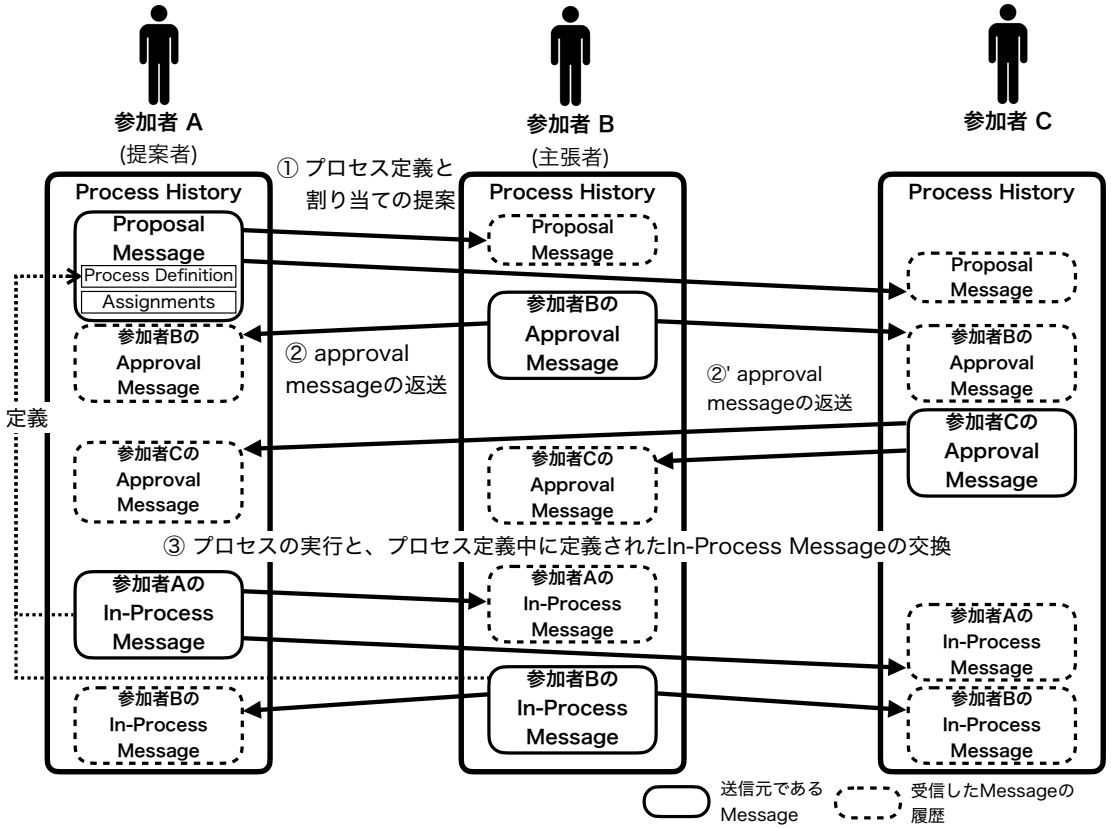

図 7.6　提案手法中の協調プロセスにおけるデータフロー: 各参加者は全てのメッセージを全参加者へブロードキャストする．これによって全参加者がメッセージを履歴として保存し，当該プロセスが実施されたことを主張可能となる．

求できる．証明書の発行要求として，Proposal，Approval，In-Process Message を含むプロセス履歴を当該実績の根拠情報として提示する．認定者は当該履歴を根拠情報として，実績を検証する．実績の検証に成功すると，認定者は当該プロセス履歴を参照する証明書を発行する．

主張者が検証者に対して実績をクレームとして主張する際は，主張者は証明書に併せて証明書発行要求で示したプロセス履歴を根拠情報として提示する．検証者は，証明書が参照するプロセス履歴が，併せて提示されたプロセス履歴であることを確認することで，認定者が当該履歴を基に実績を検証し，証明書を発行したことを確認できる．

7.4 実装

本節では，本手法の実現可能性を検討するために実装したプロトタイプを概説する．まず，本手法におけるデータモデルの詳細と，プロセス履歴の確認手順を定義する．次に，データモデルに沿った履歴を作成し，確認手順を実装し，パフォーマンスを分析する．以降の議論では，協調プロセスの定義には BPMN を使うものと仮定する．

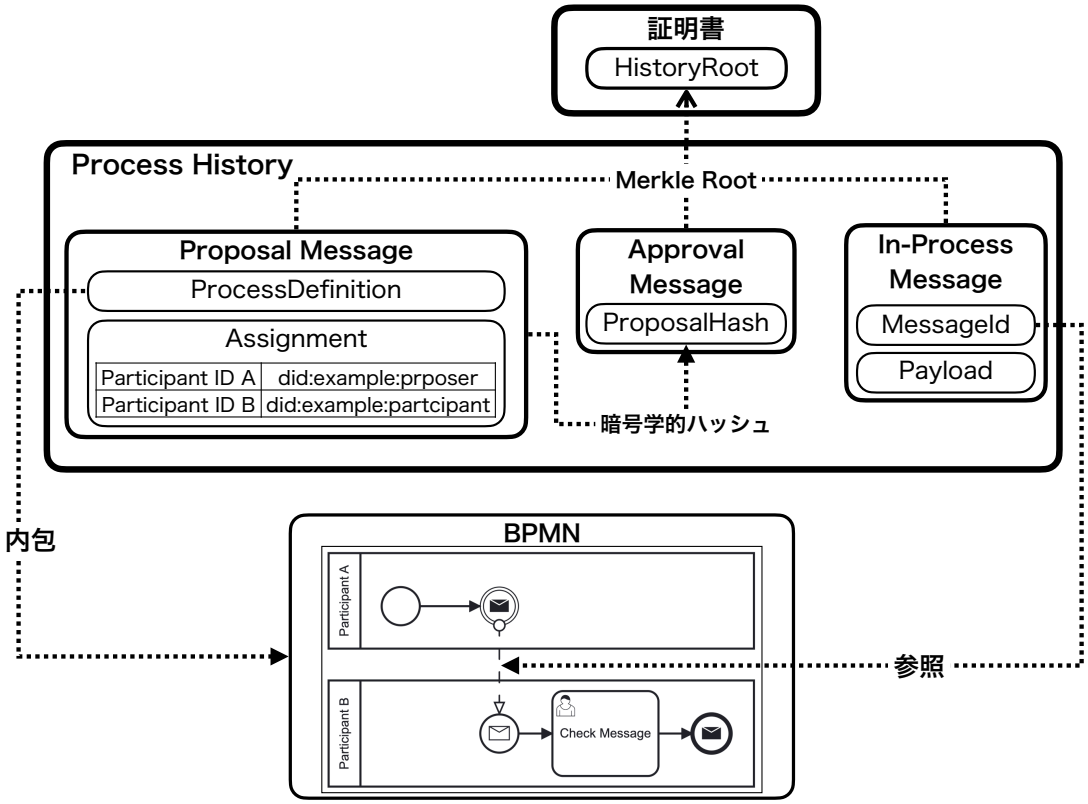

図 7.7　各メッセージの関係とデータモデルの詳細

7.4.1　データモデル

本節では各メッセージのデータモデルの詳細とそれぞれの関係を述べる．本データモデルでは，特定のプロセス定義に基づき参加者間でメッセージが交換されたこと，および当該メッセージを含むプロセス履歴に基づき証明書が発行されたことを，検証者視点で確認可能にする．図 7.7 に，データモデルの詳細を示す．

Proposal Message

Proposal Message は，参加者の一人である提案者が特定の協調プロセスの定義と，プロセス中のタスクおよびメッセージ送信の割り当てを提案することを示す．Proposal Message は以下の要素を含む．

- **ProcessDefinition**: 協調プロセスの定義のために，XML 形式にシリアライズされた BPMN
- **Assignment**: BPMN 上で各レーンに割り当てられる Participant ID と協調プロセス中の各参加者の識別子の対応を示すマップ

提案者は Proposal Message を協調プロセスの定義を示すために送信する．参加者は，当該プロセス定義と割り当てを確認し，承認するかどうかを判断する．

Approval Message

Approval Message は，特定の Proposal Message に含まれる協調プロセスの定義と，タスクおよびメッセージ送信の割り当てに参加者が合意したことを示す．Approval Message は以下の項目を含む．

- **ProposalHash**: Proposal Message の暗号学的ハッシュ値

参加者は Proposal Message への合意を示すために Approval Message を送信する． 提案者は Approval Message が Proposal Message の暗号学的ハッシュ値を含んでいることを確認する．確認できれば，各参加者は Proposal Message に合意したものとみなす．

In-Process Message

In-Process Message は，合意されたプロセス定義中に含まれるメッセージが送信されたことを示す．In-Process Message は以下の項目を含む．

- **MessageId**: プロセス定義に含まれるメッセージフローの識別子
- **Payload**: 当該メッセージで送信されるデータ

協調プロセス実施の中で，参加者は In-Process Message を交換する．MessageId フィールドがプロセス定義中のメッセージフローを参照することで，認定者と検証者は当該メッセージがプロセス定義に沿ったものであることが確認できる．

証明書

提案手法において証明書は，認定者が特定のプロセス履歴全体に基づき実績を検証し，発行されたことを示す．証明書がプロセス履歴全体を参照していることを検証可能にするため，特定のデータが木構造の中に含まれるかどうかを効率的に確認可能な Merkle 木を活用する [105]．Merkle 木の特性を活用し，証明書は以下の要素を含む．

- **HistoryRoot**: Proposal，Approval，In-Process Message を含むプロセス履歴全体の Merkle 木のルート

HistoryRoot による証明書からプロセス履歴全体の参照によって，主張者が当該履歴を認定者に対して根拠情報として提示したことを示せる．

7.4.2 履歴の確認手順

主張者が証明書の発行を要求する際，主張者は協調プロセス中に交換された全てのメッセージを，実績の根拠情報として認定者に提示する．認定者は，以下の手順でプロセス履歴を確認する．

1. 全てのメッセージの署名を検証する．
2. Proposal Message の暗号学的ハッシュ値が，Approval Message の ProposalHash フィールドの値と一致することを確認する．
3. Proposal Message 中のプロセス定義に含まれるメッセージフローに対応した In-Process Message が提示されていることを確認する．
4. 全ての In-Process Message が Proposal Message で割り当てられた参加者によって送信されていることを確認する．

以上の確認手順によって，認定者は参加者がプロセスの定義に従ってメッセージを交換し，当該プロセスを実施したことを確認できる．

主張者が検証者に当該実績を証明書を用いて主張する際は，プロセス履歴を併せて提示する．検証者は上記と同様の確認手順でプロセス履歴を確認したのち，証明書中の HistoryRoot フィールドが，当該履歴の Merkle 木のルートであることを確認する．以上の手順により，検証者は証明書の発行プロセスにおいて，認定者が当該プロセス履歴を確認した上で証明書を発行していることが確認できる．

7.4.3 データモデルおよび確認手順の実装

本ケーススタディにおける提案手法のプロトタイプを Node.js を用いて実装し，期待通り動作することを確認した．証明書は Verifiable Credentials ベースであるとした．プロセス定義としての BPMN をオープンソースの BPMN モデリングソフトウェアである Camunda Modeler を用いて作成した [106]．また，各エンティティの検証鍵と紐づいた識別子として，Decentralized Identifiers (DIDs) である did:key method を利用した [107, 108].

各メッセージのシリアライゼーションフォーマットとして JSON Web Token (JWT) を採用した [49]．それぞれのメッセージにおいて，iss クレームで送信者の識別子，aud クレームで受信者の識別子を指定する．JWT の標準では，aud クレームに指定されていないエンティティは JWT を拒否しなければならないと定められているが，本プロトタイプでは協調プロセスの履歴を確認する検証者を事前に指定することは困難であるため，この制約を除外した．より適切なシリアライゼーションフォーマットの選定は本プロトタイプ実装の範囲外とする．

表 7.1　計測におけるケース設定

Setup	参加者の数	In-Process Message の数	履歴中の JWT の数
Setup 1	3	3	7
Setup 2	3	5	9
Setup 3	5	5	11

表 7.2　計測環境

環境	設定
CPU	3.2 GHz 6 Core Intel Core i7
Memory	16 GB
Operating System	MacOS 13.6.4
Node	v16.14.0

本実装を通して，データモデルと確認手順が設計通りに動作することを確認した．

7.4.4　パフォーマンス分析

次に，プロトタイプ実装を用いてパフォーマンス分析を行った．以下の 3 つのケースにおいて，それぞれ確認手順を実行する時間を計測した．

- **Setup 1**: 協調プロセスに 3 人の参加者が参加し，3 つの In-Process Message が協調プロセス内で定義されるケース
- **Setup 2**: 協調プロセスに 3 人の参加者が参加し，5 つの In-Process Message が協調プロセス内で定義されるケース
- **Setup 3**: 協調プロセスに 5 人の参加者が参加し，5 つの In-Process Message が協調プロセス内で定義されるケース

表 7.1 にケースの設定を示す．表 7.2 に実験環境を示す．各ケースにおける履歴を作成した上で，それぞれのケースにおいて確認手順を 100 回実行し，それぞれの実行時間を計測した．

図 7.8 に実験の結果を示す． 実験の結果，In-Process Message が多くなるほど，検証するべき JWT の数が増えるため，検証にかかる時間は増加することが明らかになった．さらに，参加者の増加によっても，Approval Message の数が増加するため，同様に検証時間は増加する．以上より，参加者の数と In-Process Message の数によって，本手法における履歴の確認によるオーバヘッドが生じることが明らかになった．したがって，オー

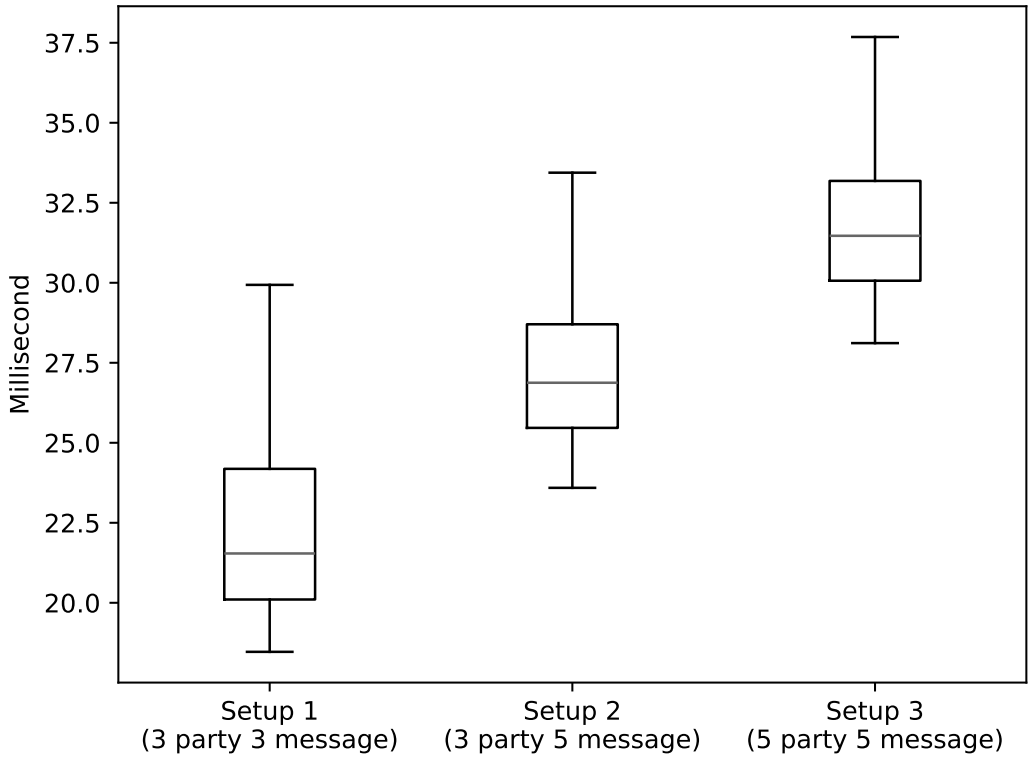

図 7.8　パフォーマンス分析の結果: 実験の結果，本手法は参加者と協調プロセス中のメッセージの数によって履歴の検証に必要な時間が増加することが明らかとなった．

バヘッドを緩和する手法の検討は本手法の今後の検討課題の 1 つである．

7.5 脅威分析と要件の充足

本節では，本手法に対する脅威分析と，要件の充足について議論する．まず，提案手法の限界を示すために想定される脅威に対して本手法の対応を議論する．次に，本手法が要件を満たすことを分析する．

7.5.1 脅威分析

実績に至る特定の協調プロセスを認定者が確認したことを，検証者視点で確認可能にするためには，協調プロセスの履歴の完全性が必要である．したがって，本手法を以下の 3 つのフェーズに分け，それぞれフェーズにおいて想定される脅威に対して本手法がどのように対応可能かを議論する．

- **協調プロセス中の脅威**: 本手法中の，協調プロセス実施中 (図 7.5 における①から

③) に想定される脅威.

- **証明書発行における脅威**: 本手法中の，証明書発行のやり取り (図 7.5 における④から⑥) に想定される脅威.
- **証明書と履歴を用いたクレーム主張/検証時における脅威**: 本手法中の，証明書と履歴を検証者に提示するやり取り (図 7.5 における⑦から⑧) における脅威.

協調プロセス中の脅威

協調プロセスの実施中，攻撃者は中間者攻撃などによって参加者間の通信に介入し，メッセージの改ざんを試みる可能性がある．しかし，全てのメッセージは署名付きデータであり，各エンティティは特定エンティティが送出したメッセージの完全性を確認可能である．したがって，メッセージの改ざんは検知可能である．また，以下のフェーズにおいても，全てのメッセージは署名付きデータであるため，中間者による改ざんは検知可能である．

証明書発行における脅威

実績を偽るために，主張者は実施していない協調プロセスの履歴を偽造し，証明書発行を要求することが考えられる．しかし，全てのメッセージは署名されているため，主張者単体では協調プロセス中の他の参加者によるメッセージの履歴を作成できない．したがって，本脅威は主張者単体では実行不可能である．

一方で，全参加者で共謀した場合，虚偽の履歴を作成する可能性がある．しかし，認定者による協調プロセスの確認を通過し，証明書の発行を受けるためには，参加者はプロセスの定義に合意し，定義に沿った形でメッセージを交換する必要がある．これは，協調プロセスを実施したことに他ならず，認定者は履歴の確認手順に基づいてその実施を確認できる．したがって，本脅威は本手法に対する脅威として成立しない．協調プロセスを認定者が確認した上で，実績を検証する具体的な検証基準は本ケーススタディの範囲外である．

認定者への証明書発行要求前に，主張者が履歴を第三者による攻撃などの理由によって喪失するケースが考えられる．しかし，全参加者は全ての履歴を保持しており，主張者は参加者から再度履歴の共有を受けることが可能である．また，主張者は共有された履歴の完全性を，各メッセージのデジタル署名によって確認可能である．

証明書と履歴を用いたクレーム主張/検証時における脅威

実績を偽るために，主張者が参加者および認定者と共謀し，協調プロセスの履歴と証明書を作成することが考えられる．しかし，前節における全参加者が共謀するケースと同様に，この脅威は協調プロセスを実施し，その履歴を確認した上で認定者が証明書を発行し

たことに他ならない．もし確認手順を通過しない履歴に対して証明書を発行している場合，主張者は検証者に履歴と証明書を提示するため，検証者は認定者が実施しているはずの確認手順によって検知できる．したがって，本脅威は本手法において対処可能である．

また，なんらかのインシデントによって主張者が履歴と証明書を喪失した場合においても，主張者は参加者から履歴の提供を受け，再度認定者に証明書発行を要求できる．また，認定者自身が発行要求を保管し，主張者から再度発行の要求を受けた場合にも，同様に証明書を再発行可能である．例えば，証明書発行者としての責任を示すモチベーションのある認定者は，履歴を安全に保管するモチベーションがあると考えられる．したがって，本脅威は対処可能である．

以上の議論により，本手法は全てのフェーズにおける脅威に対処可能であると結論づける．

7.5.2 要件の充足

本節では，7.3.2 節で定義した要件を本手法が満たしたかどうかを議論する．本手法では，主張者は証明書と併せ，参加者が合意の上実行した協調プロセスの定義を含むプロセス履歴を検証者へ提示する．したがって，検証者は Proposal Message 中の協調プロセスの定義により，実行されたプロセスに含まれるメッセージとその順序の定義を確認できるため，**R-1** は満たされている．次に，プロセス履歴に含まれる Proposal Message には，各タスクの参加者への割り当てが含まれており，いずれの参加者が協調プロセスの定義に含まれるメッセージを送信するべきかが確認できる．したがって，**R-2** は満たされている．さらに，プロセス履歴中の各メッセージには，各参加者の識別子に対応した検証鍵によって検証可能なデジタル署名が付与されている．したがって，**R-3** も満たされている．最後に，認定者は根拠情報として示されたプロセス履歴への参照を証明書に含めるため，当該履歴に基づき認定者が実績を検証した上で証明書を発行していることを検証者は確認可能である．したがって，**R-4** は満たされている．

以上により，本手法は全ての要件を満たしたと結論づける．

7.6 本ケーススタディの今後の展望

本節では，本手法の今後の展望を述べる．まずは手法全体に対して議論し，次に本ケーススタディにおける BPMN を用いた実装固有の展望を議論する．

7.6.1 ユースケースとエコシステムデザイン

本ケーススタディでは，デジタル証明書を用いた典型的なクレーム検証基準を更改するにあたり，証明書発行の根拠情報を紐づける手法を提案した．一方で，どのような協調プロセスが実績として認定され，実績たりうるかは，検証者を含めた関係エンティティが実績とみなすかどうかに依存する．また，本ケーススタディでは各エンティティに紐づく識別子には，当該エンティティによるデジタル署名が検証可能な検証鍵が紐づくことを仮定したが，対象領域によってエンティティの識別子として様々なものが想定される．例えば，学術領域では研究者を対象とする識別子である ORCID などを利用することが考えられるだろう[*1]．本ケーススタディの実装で用いた識別子である Decentralized Identifiers の標準では，識別子に紐づく情報を保管し，参照可能にするための抽象的なリポジトリとして “Verifiable Data Registry (VDR)” が定義されている [108]．しかし，VDR の実装と運用形態は各 DID method の実装に依存する．本手法を運用するためには，特定の協調プロセスを通じた実績に関係するステークホルダ間で，協調プロセスの定義への合意とガバナンス，識別子と鍵管理手法などを議論する必要があるだろう．また，先述のような識別子の管理やレジストリ運用にかかるコストを，どのエンティティがどのように分担するかも，エコシステムの重要な設計要素である．本ケーススタディで取り上げた教育以外にも具体のユースケースを検討しつつ，本手法を含む証明書エコシステムをデザインすることは，実用化に向けた重要な課題の 1 つである．

7.6.2 エンティティと識別子の紐付け

本ケーススタディでは，本手法に登場するエンティティ間で，それぞれのエンティティの識別子はお互いに確認できているものと仮定した．しかし，履歴と証明書，またそこに紐づく識別子に対応した署名鍵を主張者が第三者に提供することで，当該第三者はそれらが示す実績を主張可能である．したがって，プロセスに参加したエンティティと，主張者が同一のエンティティであることを確認する手法が必要である．デジタル証明書にまつわり，識別子とエンティティを紐付け，検証者が確認可能にするための手法が議論されている [109, 110]．これらの手法を本手法に統合し，主張者が実績を獲得した協調プロセスの参加者であることを確認可能にすることが期待される．

7.6.3 プライバシに関する議論

本手法における協調プロセス中のメッセージが参加者以外に秘匿するべき情報が含まれる場合，検証者に全体の履歴を提示することは好ましくない．この問題へ対処するた

[*1] `https://orcid.org`

めに，SD-JWT などの技術を適用しながら，情報の一部を秘匿しながら提示する選択的開示の手法を適用することが考えられる [24, 25]．また，本ケーススタディにおける実装では，履歴全体と証明書の紐付けを確認可能にするため Merkle 木を活用した．したがって，検証者に履歴を提示する際に履歴全体ではなく，履歴から構成される Merkle 木の部分木を提示することで，一部の履歴を検証者に提示することなく，当該部分木を含む履歴を認定者が確認したことを示せる．これらの手法を本手法に統合し，クレーム検証に必要な最低限の情報開示を検討することは，ユースケースおよびエコシステムデザインと共に重要な検討課題である．

7.6.4 実装に対する展望

本節では，協調プロセスの定義に BPMN を用いた本実装固有の展望を議論する．

BPMN の制約

BPMN の標準では，プロセス中に交換されるメッセージの内容を記述，定義する方法は定められていない．したがって，特定のメッセージが含むべき内容は，特定の BPMN で定義されるプロセスの参加者が合意する必要がある．BPMN を用いてプロセスを記述し，当該プロセスを実行するエンジンの実装では，BPMN の標準で定められていないメッセージの内容について独自に定義することで，メッセージの内容を記述可能にしている．BPMN が適切であるかを含め，本手法において適切なプロセス定義手法を検討する必要があるだろう．

BPMS への統合

BPMN で定義されたプロセスを実行するシステムとして，Business Process Management System (BPMS) がある．本手法を様々なプロセスに適用し，運用するためには BPMS に本手法を統合することが考えられる．一方，多くの BPMS は単一の組織内で内部のプロセスを記述，運用を想定した設計であり，複数の組織にまたがるプロセスを実行するための BPMS は議論が進行中である．本手法は，単一組織内のみならず，複数エンティティによる協調プロセスを対象とするため，複数組織にまたがるプロセスを実施，記録するシステムのデザインが必要になるだろう．

7.7 本章のまとめ

本章では，実績に対するデジタル証明書を用いたクレーム検証において，実績に至る協調プロセスの履歴を根拠情報として認定者が証明書を発行したことを確認可能にする手法を議論した．デジタル証明書をクレーム検証の根拠情報として用いる際，典型的には

検証者は “認定者が証明書発行前に実績を検証した上で証明書を発行すること” を仮定する．一方，悪意のある認定者の存在や，認定者の検証プロセスが危殆化するケースを考慮すると，当該仮定が破られる可能性がある．ここで，検証者視点で，認定者が特定の根拠情報を基に実績を検証した上で証明書を発行したことが確認できれば，先述の危殆化した証明書発行プロセスを検知できると考えられる．したがって，“証明書の発行にあたり認定者が根拠情報を確認したこと” の確認を，証明書を用いたクレームの検証基準へ追加する．本章では，協調プロセス参加者間で合意したプロセスの定義に沿った履歴を作成し，証明書発行時に認定者が確認した上で，証明書に当該履歴を紐づけて発行する手法を提案した．主張者が認定者に証明書発行要求として提示したプロセスの履歴を，検証者にも同様に提示することで，検証者は認定者が当該履歴を確認した上で証明書が発行したことを確認できる．本章のケーススタディを通じて，Shinken モデルに基づく検証基準を更改可能であることを実証した．

次章では，2 つのケーススタディを総括し，本研究で提案したクレーム検証のモデルの有用性を議論する．

第 8 章

ケーススタディ分析と Shinken モデルの評価

本章では，6 章と 7 章で議論したケーススタディを総括し，Shinken モデルの適用可能性を評価する．まず，双方のケーススタディにおいて，信頼を導入することで決定性と更改可能性のある検証基準を構成できたことを議論する．その上で，ケーススタディに共通するクレーム検証のためのモデルを整理し，他のケースにおいても応用が検討可能であることを議論する．最後に，Shinken モデルによってクレームの検証基準を明示的に構成することで，検証基準を分解，拡張，統合が可能であり，コンテキストに応じた様々な検証基準が構成可能なことを議論する．

8.1 Shinken モデル適用可能性の評価

本節では，2 つのケーススタディにおいて Shinken モデルを適用することで，決定性のある検証基準を構成し，更改可能性を確保できたことを評価する．

8.1.1 信頼を導入した検証基準の構成

6 章では，ブロックチェーンを履歴を保存する基盤かつその上で動作するスマートコントラクトは記述通りに動作することを仮定することで，取引結果の検証基準を構成した．これらの仮定に基づき，取引プロセスの定義に沿った履歴が存在することから，当該取引の結果を決定性のある検証基準で検証できる．また，6 章では，様々な取引プロセスを定義可能なフレームワークをデザインした．したがって，定義された取引プロセスにどのようなタスクが含まれるかにかかわらず，同等の検証基準で取引の結果が検証可能である．

Shinken モデルを適用したことにより，履歴と検証結果の整合も確認可能である．例えば，特定の主張者が，過去の異常終了した取引において正常終了したと主張したとして

も，取引プロセスの定義と，それに沿ったタスクの実行履歴のみがブロックチェーン上には記録されるため，虚偽のクレームであることが確認できる．

一方，仮定を置いたブロックチェーンの動作および特性に関しては，その仮定が破られることでクレーム検証の結果が偽陽性あるいは偽陰性になる可能性がある．ブロックチェーンは，各ノードの協調動作によって成立する分散システムである．また，各ノードが構成するネットワークは，誰でも参加できるオープンなネットワークである．したがって，“ブロックチェーンに参加する全てのノード” を定義することは困難であり，ブロックチェーンの特性が保証されることは決定困難である．そのため，ブロックチェーンを用いるアプリケーションの利用者は，その動作および特性を仮定し，信頼することで利用していると整理できる．

7 章では，実績を示すデジタル証明書において，認定者による当該実績の検証が危殆化しているリスクを鑑み，実績の検証に用いた根拠情報を証明書に紐づける手法を検討した．本ケーススタディは，典型的なデジタル証明書を用いたクレームの検証基準では，特定の認定者が発行する証明書の示すクレームは妥当であると仮定することから，当該仮定が破られるリスクを鑑みて検証基準を更改した試みである．一方で，根拠情報として取り上げた協調プロセスの履歴を，認定者が実績として評価したことの妥当性は，実績の内容と認定者の検証基準に依存することから，本ケーススタディの検討の範囲外とした．検証者視点では，根拠情報を元に認定者が実績と評価する基準は常に決定可能であるとは限らない．すなわち，証明書は，検証者にとって直接検証できない対象に対して，第三者である認定者を信頼することで検証可能性を確保するための手段である．そこで，本ケーススタディでは，認定者が根拠情報を確認しているのであれば，証明書で示される実績を認定者が妥当に検証していることを仮定して，検証者は実績の検証基準を構成している，と整理できる．

以上より，Shinken モデルを適用し，信頼を導入することで，それぞれのケースにおいて決定性のある検証基準を構成し，情報システムとしてクレーム検証を実現できた，と結論づける．

8.1.2 更改可能性の検討

6 章では，ブロックチェーンの動作および特性を信頼し，仮定することで，検証基準を構成した．一方で，仮定が破られるリスクを鑑みて，その特性が保証されることを仮定せず，ブロックチェーンが正常に動作していることを確認するよう検証基準を拡張し，更改できる．例えば，ブロックチェーン自体はその内部で各ノードの持つ台帳が一貫性を持つよう Proof-of-Work などのコンセンサスアルゴリズムを導入している．Proof-of-Work では，ブロックチェーンのネットワークに参加するノードの 51% 以上の計算パワーを握

ることで，任意の履歴を台帳に含めないよう恣意的に記録を操作できる “51% 攻撃” などが知られている．また，利用するブロックチェーンの実装に，任意の履歴を書き換え可能な脆弱性が存在する可能性は，否定できない．これらのケースにおいて，ブロックチェーン上に記録された履歴が書き換えられた場合，任意の取引プロセスの結果を書き換えられる可能性がある．取引結果の検証においてこれらのリスクを勘案する場合，ブロックチェーンが正常に動作しており，履歴の改竄が起こらない状況であることの確認を検証基準に追加することが考えられる．例えば，Proof-of-Work ベースのブロックチェーンを活用する場合，ネットワーク上の統計情報を参照することで，51% 攻撃が発生していないことを確認することが考えられるだろう．

また，様々な取引プロセスを定義可能なフレームワークとして設計したことも，更改可能性の担保に寄与している．6 章で構成した検証基準では，ブロックチェーンの動作を仮定した上で，取引プロセスは定義した通りにのみ状態遷移することから，取引の終了状態を演繹的に推論する．しかし，特定の取引プロセスで用いられる Fair Exchange プロトコルに脆弱性が発見された場合，取引の終了状態がタスクの実行結果と一致しない状況に陥ることが考えられる．この時，検証基準として構成した推論は健全でなくなる．その場合，フレームワーク上に実装された Fair Exchange プロトコルの脆弱性を修正することで，推論が健全となるように改修することが考えられる．これは，取引プロセスに必要なタスクを柔軟に定義可能な形でフレームワークを設計したことにより，検証基準の礎となる状態遷移を更改可能にしていると捉えられる．そのため，6 章では，決定性のある検証基準を構成し，様々な取引プロセスを定義可能なフレームワークとして設計したことによって，更改可能性が確保されていると評価できる．

7 章では，ケーススタディそのものが検証基準を更改する試みであり，その実現可能性を示した．本ケーススタディでの提案は，特定の認定者が発行した証明書の示すクレームは妥当であるという仮定の元に構成される典型的な検証基準に対して，実績に至る協調プロセス履歴を認定者が確認したことを検証するよう基準を拡張した．

また，Shinken モデルを導入したことにより，証明書発行時に確認した根拠情報があるにもかかわらず，妥当とはみなせないクレームが証明書に含まれていた場合，認定者の根拠情報を用いたクレーム検証が妥当に行われていないことが演繹的に推論できる．さらに，認定者が妥当にクレームを検証しないケースを考慮する場合は，認定者のクレーム検証自体の妥当性を何らかの方法で確認する基準を検証基準に追加することが考えられる．また，検証基準の更改をせずに，当該認定者が妥当なクレーム検証を行なっていないことを指摘し，情報システムの外部で責任を追及することも運用上は可能である．実際に検証基準を更改するかどうかは，検証者自身が情報システムによるクレーム検証にどこまで依存し，検証結果をどのように取り扱うかという運用上の観点の議論が必要となるだろう．

以上 2 つのケーススタディより，Shinken モデルを適用することで，更改可能な検証基

準を構成できたと評価する．また，前節の議論と併せ，Shinken モデルはそれぞれのケーススタディにおいて適用可能であり，その有用性が確認できたと結論づける．

8.2 ケーススタディに共通するクレーム検証構成の抽出

本節では，2 つのケーススタディに共通するクレーム検証の構成を整理し，本研究のケーススタディが他のケースにおいても応用が検討可能であることを議論する．

8.2.1 協調プロセスとその結果の検証基準の構成

両ケーススタディでは，協調プロセスの結果を検証するために，その履歴を活用する検証基準を構成した．協調プロセスは複数エンティティが参加し，複数のタスクから構成されるプロセスである．その結果は，当該協調プロセスの実態が何であるかに依存するが，どのようなタスクで構成され，どのような状態遷移を辿るかが定義されれば，各タスクの実行履歴から “特定の終了状態に遷移したこと” の検証基準が構成できる．そこで 6 章では，協調プロセスとして取引プロセスを定義し，当該取引プロセスの終了状態を検証可能な手法をデザインした．また，7 章では，デジタル証明書発行に至る確認の対象として，協調プロセスの履歴を示すデータモデルをデザインした．

両ケーススタディでは，協調プロセスに含まれるタスクを定義し，それに沿った履歴を作成することが共通点として挙げられる．また，検証者はこれらの履歴を元に結果の検証基準を構成し，クレーム検証を実現する．このことから，多様な検証者が特定の協調プロセスの結果の検証基準を成立させるために，以下の要素が必要である．

- 形式化されたタスクの定義と，定義に沿った履歴のフォーマット（協調プロセスのモデル化）
- 履歴を用いて当該協調プロセスの結果を検証するエコシステム

前者は，協調プロセスの履歴を情報システムで処理するために必要となる．後者は，多様な検証者が同様の履歴を根拠情報としてクレーム検証を実施するために，それぞれで同様の検証基準を共有する必要があることを示している．つまり，当該協調プロセスの結果を主張あるいは検証するエンティティによって構成されるコミュニティ内で，特定の履歴フォーマットを用いることを合意し，相互運用性を確保することが必要である．このエコシステムの中では，特定の協調プロセスが何のタスクで構成され，その結果が実績などとして取り扱えることを合意する．例えば，特定の講義のプロセスを実施した結果，特定の教育効果が得られ，特定のスキルを受講者が獲得できることを合意する時，受講者は特定のスキルの獲得をクレームとし，講義のプロセス履歴を根拠情報として提示できるだろう．

8.2.2 考えられる応用先

協調プロセスは人々が生活する中の様々な場面に存在するため，本検証基準を応用可能な対象は多岐にわたる．本節では，いくつかの考えられる応用先を概説する．

サプライチェーンの透明性確保

サプライチェーンでは，製品が製造，出荷，流通，販売という複数のプロセスを経て製品が消費者へと流通していく．この中で，製品に関連する情報を検証可能にするため，協調プロセスとしてサプライチェーンの一連のプロセスをモデル化し，履歴を活用することが考えられる．IETF の Supply Chain Integrity, Transparency, and Trust (scitt) working group では，サプライチェーンの信頼性と透明性を確保するアーキテクチャを議論している [111]．このアーキテクチャでは，サプライチェーンに関係するエンティティは，製品に関わる情報を署名付きデータである Signed Statement として記録し，監査可能にする．本研究で議論したモデルを適用することで，記録された Signed Statement を根拠情報，製品に関連する情報をクレームとして検証することが考えられるだろう．

組織へのオンボーディングプロセスの透明性確保

あるエンティティが組織へオンボーディングする際は，当該エンティティの身元確認を含め，複数のプロセスを経る．中でも，オンボーディングプロセスの結果，当該組織に所属するエンティティのアイデンティティを管理するアイデンティティプロバイダ (IdP) と呼ばれるシステムへ，対象エンティティの情報が記録される．当該アイデンティティは，組織の内部のみならず，他組織との連携においても OpenID Connect などのプロトコルを用いて，フェデレーションされるケースがある [86]．この時，オンボーディングのプロセスにおいて，どのような本人確認書類の提出によって身元確認あるいは当人認証したかの情報は，アイデンティティをクレームとして検証する際にその妥当性を向上させる根拠情報になると考えられる．したがって，オンボーディングのプロセスをモデル化し，履歴を活用することで，より確度高い形でアイデンティティの検証を実現することが期待できる．

学術論文査読プロセスの透明性確保

学術論文の査読は，論文の妥当性を査読者が判断し，論文誌に掲載される学術論文の質を担保するための重要なプロセスである．一方で，査読プロセスの透明性と質に関して様々な問題が指摘されている [112]．例えば，査読者によって同一の論文の査読結果にランダム性があることや，査読記録が有効に活用されず，論文が再投稿された際に効率化できないこと，査読者が匿名であるが故に十分に論文の内容を精査しない，あるいは査読を

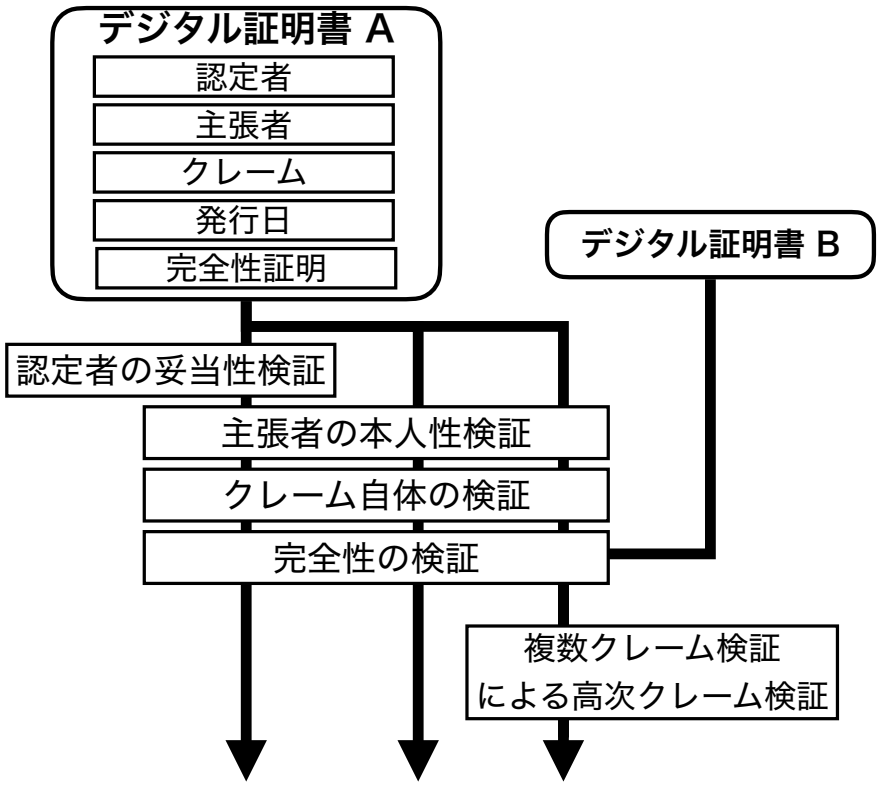

図 8.1　Shinken モデルに基づいたクレームの検証基準の分解，拡張，統合: 単一の証明書には複数のクレームが含まれることから，個々のクレームに対する検証の組み合わせによって検証者が必要とする検証基準が成立する．Shinken モデルでは，個々のクレームに対する検証基準は決定性があるため，独立して成立し，分解が可能である．また，拡張の一形態として，異なる証明書を組み合わせたクレームに対しても，決定性のある検証基準を構成可能である．

引き受けたにもかかわらず遅延する査読者の存在が挙げられる．査読は，著者，編集委員，査読者による協調プロセスであることから，プロセスをモデル化し，履歴を活用することでそのプロセスの透明性を確保することが考えられる．例えば，査読者は多くの場合ボランティアであることから，査読に参加するインセンティブとなるように査読への参加自体を研究者の実績として取り扱う議論がある．この場合，査読プロセスの履歴は査読へ参加したことの根拠情報として活用することが期待できるだろう．

8.3 クレームの検証基準の分解，拡張，統合

本節では，Shinken モデルによって，クレームの検証基準を明示的に構成することで，基準を分解，拡張，統合できることにより，様々なクレームの検証を検証者のコンテキストに合わせて実現可能であることを議論する．

図 8.1 に，Shinken モデルに基づいたクレームの検証基準の分解，拡張，統合の概要を示す．まず，単一のデジタル証明書に複数のクレームが記載されるケースを考える．例えば，学位証明書は，特定のエンティティが特定の学位を取得したことを示すために，以下の情報が含まれる．

- 認定者である大学の名称などの情報
- 対象エンティティを示す氏名などの情報
- 取得学位とその名称
- 取得日
- データの完全性を示すデジタル署名などの情報

この時，これらの情報の一部を選択的開示の手法を適用し秘匿することで，デジタル証明書として 1 つのデータで示される情報を分解し，部分的に提示できる．一方で，特定のエンティティが学位を取得していることをクレームとする時，検証者による検証基準は，以下のような複数のクレーム検証から成立する．

認定者の検証 認定者が当該学位を与えるに足る妥当なエンティティであることの検証．学位の場合，日本における文部科学省など，適切な機関によって学位認定機関であると認定されていることを確認することなどが含まれる．

対象エンティティの検証 提示された証明書の示すエンティティと，主張者が同一のエンティティであることの検証．

取得学位の検証 証明書の示す学位が妥当な学位であることの検証．

デジタルデータとしての完全性検証 デジタル署名などによってデジタル証明書の完全性の検証．

これらの検証は Shinken モデルではそれぞれ独立した決定性のある検証基準で構成する．このとき，上記の検証基準の一部は，検証者の意図など，コンテキストよっては無視できる．例えば，認定者が検証者にとって既知であり，妥当な認定者であることが仮定できる場合，認定者の検証をせずに，その他の項目を検証することで，目的のクレーム検証を実現できたと見なせる．したがって，Shinken モデルの導入によって，個々の検証基準は分離できるため，デジタル証明書で示される情報を分解し，検証者の必要とする最低限の情報のみの提示によってクレーム検証を実現できる．

また，Shinken モデルに基づく個々の検証基準は決定性をもつため，既存の検証基準に対して別の決定性のある基準を必要に応じて追加し，拡張できる．例えば，取得学位の検証において，特定の学位のみを受け入れるような検証基準を追加することが考えられる．この時，検証結果に対して追加した検証基準も決定性と透明性をもつため，影響が明確で

ある．

拡張の一形態として，特定のクレームの検証基準を分解し，他のクレームに対する検証基準の拡張として統合可能である．例えば，複数の学位の取得を繰り返し上記の検証基準を用いて確認しつつ，主張者の一連の学習歴をクレームとして検証できる．あるいは，特定の学部での学位を取得したことと，別の資格を取得していることを特定の証明書を用いて検証する基準を統合することで，“主張者が特定の知識を一定以上持っている”ことの検証基準を構成できるだろう．このように，複数のクレーム検証を組み合わせ，個々のクレームが相互に根拠情報となる検証基準を構成することで，多様なクレームの検証を実現できる．これによって，特定の単一の証明書が示す内容を超えて，複数の証明書を組み合わせて統合された情報を活用しながら，クレーム検証を実現できる．

以上の議論により，本研究で提案した Shinken モデルに基づき，様々な検証基準を分解，拡張，統合し，組み合わせながら，コンテキストに合わせたクレーム検証を実現可能と結論づける．

8.4 本章のまとめ

本章では，2 つのケーススタディを総括し，Shinken モデルの適用可能性を評価した．まず，それぞれのケースにおいて信頼を導入し，決定性のある検証基準を構成できたことを分析した．6 章のケーススタディ 1 では，ブロックチェーンの動作および特性を信頼し，7 章のケーススタディ 2 では実績に至る協調プロセスの履歴を認定者が確認した上で，適切に評価したことを信頼することでそれぞれの検証基準を構成した．また，信頼する仮定が破られたケースにおいて，それぞれ仮定した部分を更改することでクレームをより確度高く検証できることをそれぞれのケースにおいて分析した．以上により，Shinken モデルがそれぞれのケーススタディにおいて適用可能であったと結論づけた．次に，両ケーススタディに共通する協調プロセスの結果をクレームとして検証するモデルを整理した．その上で，同様の検証基準が様々なケースにおいて応用を検討可能であることを示した．また，Shinken モデルを適用することで，情報システムとしてクレームの検証基準を分解，拡張，統合可能であり，コンテキストに合わせたクレーム検証を実現可能であると結論づけた．

第 9 章

結論

本章では，本研究の結論を述べる． まず，本研究の議論を総括する．その上で，本研究に基づく新たな研究の可能性を述べ，本研究の意義をまとめる．

9.1 本研究の総括

本研究では，インターネットを通じた情報のやり取りの中で，クレームの妥当性検証に着目した．まず，インターネットを通じて人々が協調する活動を協調プロセスとして整理した．その上で，個人の実績の検証や，多様化するシステムへのアクセスコントロール，フェイクニュースなどの観点から，情報の検証可能性が議論されていることを指摘した．また，日本語の検証にあたる英語の概念として，検証者による対象の確認にかかる意図や目的などのコンテキストを尊重し，妥当性を確認する Validation と，コンテキストによらない形式を確認する Verification があると整理した．情報の検証可能性にまつわる議論の中で活用されるデジタル証明書の標準では，デジタル署名をはじめとしたデータセキュリティ技術の適用による Verification は対象とするものの，Validation は検討の範囲外とされている．そこで，本研究は Validation に着目し，情報の妥当性検証を議論した．

一方で，デジタル証明書の活用の目的の多くは，デジタル証明書に示される “クレーム” をアプリケーションの中で活用することにある．しかし，先述のように，クレームの妥当性検証はデジタル証明書の標準では対象外とされ，クレームの妥当性検証を情報システムで実現するためのモデルの議論は不十分であった．そのため，典型的には，特定のデジタル証明書に示されるクレームの妥当性検証の基準は，そのコンテキストに依存した形で構成され，当該コンテキスト特有の仮定や条件が前提として基準に組み込まれる．したがって，異なるコンテキストで特定の証明書を活用する場合，元々の前提を暗黙的に援用してしまい，検証者の目的に応じた適切な検証基準を構成できない可能性がある．この問題に対処するため，クレームの検証者の視点で，コンテキストに応じた妥当性検証

の基準を構成するモデルが必要であることを提起した．

そこで，本研究では，デジタルデータで示される複数の根拠情報を活用しながら，検証者のコンテキストに合わせてクレームの妥当性検証を実現するためのモデルを提案した．まず，デジタルデータに非依存なやり取りのモデルとして，クレーム検証をモデル化し，その概念とエンティティの関係性を整理した．本研究において，クレーム検証とは，主張者が提示するクレームと根拠情報が，検証者の検証ポリシで示される検証基準へ適合するかどうかを判断することであると整理した．その上で，証明書を用いた特定のクレームに対する妥当性検証の成立過程を整理した上で，やり取りのモデルを整理した．主張者，検証者に加えて，証明書発行者である認定者がクレームを検証した上で，検証したことを示す証明書を発行することで，検証者が直接検証できないクレームに対しても，妥当性検証を実現できることを示した．また，証明書を用いたクレームの妥当性検証は検証者の検証ポリシを起点とする，各エンティティのポリシの参照関係によって実現される．したがって，検証者の検証ポリシで示される検証基準への適合あるいは非適合を判断する判定問題として整理できる．

そして，情報システムでクレーム検証を実現するために，クレームと根拠情報を入力とし，検証者の検証ポリシに示される検証基準への適合あるいは非適合を出力する抽象的なインターフェースの定義として，`Validation` 関数のモデルを定義した．情報システムでクレーム検証を実現することは，検証ポリシに合わせて `Validation` 関数を検証器として実装することである．この時，検証器に実装される検証基準は判定問題であることから，形式論理に基づくものと，確率論に基づく実装が考えられる．検証器を実装以降，検証対象のクレームを取り巻く検証者内外の要因により，検証ポリシは変化しうることを考慮すると，検証器に実装される検証基準はポリシの変更に追従して更改可能であることが好ましい．また，更改にあたっては，その更改前後の差異を特定可能であることが望ましいことを指摘した．

その上で，更改可能な検証基準の構成のための抽象モデルとして，演繹的推論に基づく検証基準を構成する “Shinken モデル” を提案した．クレーム検証の特性から，検証器の入力に対する検証結果が決定論的である “決定性” と，検証器の結果が当該の結果に至った要因が特定できる “透明性” を要件として定義した．2 つの性質を満たすためには，形式論理を基に検証基準を構成することが望ましいが，検証基準には計算機で決定不可能な命題を含む可能性がある．そこで，Shinken モデルでは検証器に実装される検証基準の中で一部の命題を真であると仮定することで，基準全体の決定性を担保する．また，一部の命題を真であると仮定することをクレーム検証における “信頼” であると整理した．Shinken モデルに基づいた検証基準の構成例として，デジタル署名の活用を示した．真であれば署名を付与する命題を “署名の意図” と整理し，デジタル署名が検証できることから，署名の意図を演繹的に推論することで，クレーム検証を実現できる．

次に，Shinken モデルに基づき要件を満たしながらクレームの妥当性検証を実現できることを示すため，ケーススタディを 2 件議論した．1 つ目のケーススタディでは，商取引において，取引相手の不誠実な振る舞いによる経済的損失の回避のために，当該取引相手の過去の取引結果をクレームとして検証するケースを取り上げた．取引に含まれるタスクの定義と，定義に沿った履歴を活用することで，取引結果の検証基準を構成した．また，取引の定義および履歴が改ざん困難な形式で記録されることを仮定するために，ブロックチェーン技術を適用した．構成した検証基準に基づき，取引の履歴を記録し，根拠情報として活用するために，ブロックチェーン上で様々な取引を定義しその履歴を参照可能な取引フレームワークを設計した．本フレームワーク上で実施された取引の結果は，ブロックチェーン上に記録された取引の定義と履歴によって検証可能である．

次に，2 つ目のケーススタディとして，デジタル証明書に示されるクレームの検証基準の更改を検討した．デジタル証明書を活用したクレーム検証では，その発行元である認定者がクレームを検証したことを署名の意図として仮定した上で，検証基準を構成する．しかし，証明書単体では認定者が真に検証した上で発行したかは確認できず，当該仮定が破られた場合，クレームを誤認する可能性がある．ここで，クレームの検証基準を更改し，“認定者が特定の根拠情報を確認の上で証明書を発行したこと” を検証することで，当該仮定が破られるような発行プロセスの危殆化を検知可能になると考えた．そこで，証明書の発行にあたり認定者が確認した根拠情報を証明書に紐づいた形で発行する手法を提案した．特定の協調プロセスを通じて主張者が獲得する実績をクレームとする証明書を対象に，根拠情報として当該協調プロセスの履歴を紐づける手法を提案した．ビジネスプロセスを機械可読な形式で記述する BPMN を活用しながら，協調プロセスの履歴のデータモデルおよび検証手順を定義し，プロトタイプ実装によって期待通り動作することを確認した．

Shinken モデルの適用可能性の評価として，信頼の導入により決定性と更改可能性のある検証基準が構成できたことを 2 つのケーススタディを総括しながら分析した．また，2 つのケーススタディに共通する検証基準を抽出し，本研究で議論した適用先以外にもサプライチェーンや組織へのオンボーディング，学術論文の査読など多様なプロセスに対して応用が検討可能であることを示した．最後に，情報システムとして決定性のあるクレームの検証基準を構成することで，その基準が分解，拡張，統合できることをデジタル証明書の活用を対象に議論した．以上により，情報システムで複数の根拠情報を活用しながら，コンテキストに応じたクレームの妥当性検証を実現するモデルが確立できたと結論づけた．

9.2 本研究に基づく新たな研究の可能性

本節では，本研究に基づく新たな研究の可能性を述べる．まず，本研究で着目した “信頼” と “妥当性検証” に関するモデルの理論的な発展について述べる．また，本研究で提案したモデルを，実社会の課題に対して適用する取り組みの可能性と，具体的な取り組みへと発展させるための課題を述べる．最後に，本研究と他分野の融合による，学際的な取り組みの可能性をまとめる．

9.2.1 “信頼” と “妥当性検証” の理論的深化

本項では，本研究で提案した Shinken モデルを発展させる，“信頼” と “妥当性検証” の理論的な発展の方向性を述べる．

Shinken モデルの形式的記述と形式検証

本研究では，クレームの妥当性検証を情報システムで実現するために，検証者の検証基準を示した検証ポリシを元に検証器を実装するモデルを提案した．検証ポリシは，検証者およびクレームを取り囲む環境に影響され，その際は検証基準の更改が必要になる．この時，検証ポリシと検証器に実装された検証基準の対応を確認することが重要である．システムの動作を形式的に記述することによって，その動作が意図した通りであるか形式的に検証する手法が検討されている．このような手法を援用しながら，アクセス制御やロボット制御の文脈などで，いくつかのポリシ記述言語が提案されている [113, 114]．また，そのために検証ポリシを宣言的に記述するためのポリシ技術言語を開発することも考えられるだろう．ポリシ記述言語を用いて，形式的に記述された検証ポリシと検証基準の対応を検証する手法の検討が考えられる．本研究で提案された Shinken モデルに基づき，検証ポリシを形式的に記述し，モジュール化された検証基準の分解，統合を可能とすることは，多様な文脈で柔軟性のあるクレーム検証を実現するシステム構築を支える基盤となるだろう．

また，Shinken モデルにおいては，検証器に実装される検証基準を演繹的推論を基に構成することを提案した．そこで，付録 A では，本研究の中で構成した検証基準の Prolog によって形式的な記述を示した．本研究を基に，今後の取り組みとして，検証ポリシの形式化と共に，既存の形式論理体系の適用が考えられる．本研究の提案に基づき，既存の形式論理体系に基づき “信頼” と “妥当性検証” の基準を記述することで，より多様な検証基準の構成が検討できるだろう．これらの取り組みを通じて，より実社会における複雑な情報の妥当性検証を情報システムとして構築していくことが期待できる．

“信頼すること” の妥当性検証

本研究では，クレームの検証基準の中で信頼を導入することで，情報システムとしてクレームの妥当性検証を実現できることを述べた．一方，検証基準の中で何を信頼するか，自体も検証者自身の判断である．すなわち，“信頼の対象が信頼するに足りうること” の妥当性検証が必要である．そこで，今後，本研究の提案に基づき，検証基準の中で対象が信頼するに足りうることを Shinken モデルを用いて再帰的に検証することが考えられるだろう．

例えば，インターネットを通じたサイバー空間中で人々が協調する際に，相対するエンティティが安心してやり取りを実施しうるかどうかは，大きな課題である．本研究の 6 章でも，商取引の文脈において過去の履歴を元にエンティティの信頼性を判断する手法を検討した．エンティティの信頼性を測るためには，なんらかの方法で，やり取りの当事者同士でエンティティを評価する必要がある．この時，本研究で議論したクレーム検証は，評価のための根拠として様々な情報を統合するためのモデルであるとも捉えられれる．クレーム検証の実現にあたり，信頼を導入した検証基準の中で仮定する対象に対しても，なんらかの方法で評価し，仮定を置けることを検証者は判断しなければならない．したがって，本研究の提案を，エンティティの信頼性評価を実現する検証基準の構成へと再帰的に適用することが考えられる．すなわち，様々なユースケースの中で本研究の提案を活用するためには，個々の検証基準の中で信頼し，仮定する対象の妥当性を評価する手法を検討する必要があるだろう．これらの手法の検討によって，サイバー空間中における，より安心できる情報のやり取りが実現可能であると期待できる．

9.2.2 実社会の課題への適用

本項では，実社会の具体的な課題への本研究の適用の可能性を述べる．

偽情報対策システムの構成

大規模言語モデルをはじめとした生成 AI の登場を受け，インターネット上で流通する偽情報あるいは悪意のある情報の対策は，喫緊の課題である [5]．これらの対策のため，根拠情報を構造化して保管し，判断を支援する試みが議論されている [115]．本研究では，複数の根拠情報を元に判断をするシステムを構成するモデルを提案した．構造化された根拠情報を元に，情報の受け取り手のポリシに併せて判断するシステムを構成することで，対象の情報を偽情報と判断するシステムの構成が期待される．

証明書の相互運用性の確保

学位証明書をはじめとした証明書のデジタル化と国や地域などの複数のコミュニティを跨いだ相互運用が盛んに議論されている [31]．本研究で議論したように，証明書を用いて当該クレームを検証する検証者は，証明書発行者である認定者が妥当な検証方法を定義していることを基に，証明書を確認を以て当該クレームを検証したとみなすことを検証ポリシとして定める．つまり，特定の認定者の検証ポリシが，特定のコミュニティ内で妥当であると合意されれば，当該コミュニティ内の検証者は自身の検証ポリシに当該認定者のポリシを内包し，証明書を用いてクレームを検証できる．一方，当該コミュニティ以外の別のコミュニティに属する検証者は，当該コミュニティ内部で発行された証明書を用いてクレームを検証できない．そのため，証明書の相互運用性の確保のためには，両コミュニティでそれぞれに属する認定者の検証ポリシに対して合意する必要がある．ここで，合意するためには，特定の検証ポリシが両コミュニティで合意可能である受容可能性，および合意の上でそれぞれの検証者の検証ポリシに組み込み可能なモジュール性が必要である．本研究で提案したモデルに基づき，決定性のある検証基準で認定者がクレームを検証することで，当該検証基準が妥当であるかの判断を容易にし，検証者の検証ポリシへの組み込みを可能にするモジュール性を確保することが期待できる．これらの実現のためには，ポリシ記述言語などを用いてポリシ記述を標準化することが考えられるだろう．特定のアプリケーションおよびコミュニティを超え，証明書で示されるクレームを多様な場面で再利用するためには，本研究のモデルに基づき，証明書の相互運用性を確保することが考えられる [21, 22, 23]．

9.2.3 個別具体の事例への適用に向けた課題

前節で述べた事例を含め，個別具体の事例への適用に向けた本研究の課題を述べる．本研究では，検証者の検証ポリシを起点として，検証者による明示的な検証基準を構成するための手法を提案した．一方，検証基準を構成する際は，対象のクレームがどのように主張されているか，あるいは利用できる根拠情報は何かが重要である．また，場合によっては検証者が必要とする十分な根拠情報が存在しないことが考えられる．したがって，実社会への適用に向けては，検証ポリシによる検証基準の明示化と共に，根拠情報の提供手法も併せて検討する必要がある．このような検討の中では，本研究に基づき明示的な基準を構成することで，必要とされる根拠情報が何かが明示されることが期待できる．例えば，特定の検証者にとって既存の提供可能な根拠情報が不十分である場合，十分な根拠情報をどのように提供できるかの検討，あるいは提示しうる根拠情報をもとに十分とみなす基準をコンテキストに応じて設計することが考えられる．すなわち，本研究に基づいた情報システムを実社会で運用するためには，個別具体の対象に対してコン

テキストに応じた検証ポリシと主張ポリシのネゴシエーションおよび合意のための検討が必要となる．この検討のためには，既存のシステムで暗黙的であった検証基準を本研究のモデルに基づき明示的に規定することで，その不十分となる部分を明らかにすることが期待できる．また，異なる検証者が定める検証基準同士が矛盾するようなケースも，検証基準を明示的に定めることで発見が容易になることが期待される．これらの観点から，本研究で確立したモデルは，様々なクレームを検証可能とするための，関連するステイクホルダによる検討の礎となるだろう．

また，本研究では，検証者が直接クレームを検証できないケースへの対応として，証明書を用いたやり取りのモデルを整理した．検証者がクレームを検証するためには，検証者が必要とする根拠情報を提示する必要がある．一方で，根拠情報となる証明書には複数の情報が含まれ，必要以上の情報を提示してしまうケースも考えられる．本研究では，デジタル証明書に関連した技術として，証明書の一部を部分的に開示する選択的開示の手法について議論した．今後，個別具体の事例に適用するにあたっては，検証者が検証ポリシに定める必要十分な根拠情報を，適切に提示する手法を検討する必要があるだろう．

これらの具体的な事例への適用では，本研究で着目したクレーム検証に付随した非機能要件の議論も重要である．例えば，本研究の 6 章と 7 章のケーススタディにおいては，それぞれの実装のスケーラビリティを確認するため，パフォーマンスを評価した．パフォーマンスの観点で，検証者がクレームを検証する個別具体のコンテキストに応じて，許容できる範囲に収まることが好ましい．個別具体の対象への適用に併せ，実装の側面からも本研究を発展させていくことが期待される．

9.2.4 学際的な取り組みへの結合

最後に，本研究の取り組みを起点に，他分野との結合により学際的な取り組みへの発展可能性を述べる．

セマンティックコミュニケーション

本研究で議論したクレーム検証は，デジタル証明書をはじめとしたデータが示すクレームの示す意味論を含めた情報通信モデルの一種であると捉えられる．意味論を含めた情報通信の議論として，セマンティックコミュニケーションと呼ばれる通信モデルの議論がある [116, 117, 118]．セマンティックコミュニケーションの議論では，特定のデータが示す意味などを内包する知識ベースを共有したコミュニティ内でのやり取りのモデルなどが議論されている．一方，データとして表現され，通信される情報の意味解釈は，不確実性を含む人間の認知を含む．したがって，これらの議論との関係を整理することによって，Shinken モデルにおける信頼の導入によって不確実性を縮減させ，決定性のある

情報通信モデルの議論が可能であると期待できる．これらの検討が進むとともに，データの通信を超えた，人間同士が情報システムの支援を受けながら，情報通信を実現することが期待される．

既存の “信頼” の概念との接続

本研究が扱う “信頼” の概念は，情報システムの視点のみならず，哲学，社会学，教育学など多くの学問分野で様々な視点から論じられてきた [26]．本研究は，情報システムの構成における視点から “信頼” の概念構築を試み，既存の議論に基づき “信頼される対象” を明示しながら情報システムを設計する方法論を示した．信頼される対象の選定など，本研究に基づく情報システムを活用する社会システムの観点からは，情報システムにとどまらない，既存の議論との結合による学際的な議論が必要である．本研究は，そうした学際的な議論に向けた，情報システム観点での “信頼” の再解釈と応用の可能性を開くものである．

9.3 本研究の意義

本研究では，情報システムにおけるクレームの妥当性検証をモデル化し，コンテキストに応じて妥当性判断の検証基準を決定論的な問題として構成する枠組みである Shinken モデルを提示した．これまでのデジタル証明書の標準規格などにおいて妥当性検証は範囲外とされてきた故に，証明書に示されるクレームの妥当性検証において暗黙的な仮定，即ち信頼が置かれてしまうことで，クレームの誤認が生じる可能性を孕んでいた．例えば，既存の PKI では，認定者にあたる証明書発行者を信頼することを前提とするため，危殆化した認定者によって発行された証明書を用いたことによるインシデントの事例がある．また，その証明されるクレームも “対象エンティティと検証鍵の紐付き” を中心として限定的である．そのため，証明書の利用者である検証者視点で何を検証するために証明書を使えるのか，その証明書をどう解釈するべきかを整理するモデルは存在しなかった．そのような状況を受け，DV 証明書を用いた https 通信ができるのみで “フィッシングサイトではない” という曖昧かつ確度の低い基準を暗黙的に検証者が受け入れ，リスクにさらされる状況も散見されていた [119]．また，Verifiable Credentials の標準でも，特定の発行者から発行された証明書を特定の提示者が提示した，ということを検証可能にするのみであり，検証者の視点での証明書の示すクレームの検証に踏み込むモデルは存在しなかった．そのため，学位証明書の発行者である “大学” という組織の分類が同一でも，異なる教育制度の国同士で証明書を解釈できない，証明書の相互運用性の問題が存在した．

そこで，本研究では，証明書に基づかない単純なやり取りのモデル（基本モデル）を起

点とし，証明書の役割を再定義した上で，証明書の示すクレームに対して明示的な基準で検証するモデル（証明書モデル）を提案した．中でも，Shinken モデルは，これまで暗黙的に情報システムを構成する中で処理されてきた仮定を，検証者の検証ポリシの中で信頼として明示的に導入する枠組みであり，クレーム検証を処理する情報システム設計を可能にする手法である．本研究で提案したモデルでは，証明書の示すクレームの解釈が，その発行基準をはじめとして明示的な基準に基づくため，検証者のコンテキストに応じてそれぞれの基準が適切かどうかを精査可能である．すなわち，証明書を用いたあるクレームの妥当性検証において，コンテキストに応じて誤認のリスクを考慮した上で，確認するべき事項に抜けがないかを基準全体を精査することが可能になる．また，教育制度が異なることは，証明書の発行に至る基準が異なることであると整理できる．したがって，異なる基準のもとに発行された証明書から何を解釈できるのか，検証者自身のコンテキストに併せて基準に足りない点はどこかを明示的に精査することが可能となる．

さらに本研究のモデルは，資格や学位といった制度的な実績に限らず，これまで検証が困難とされてきたプロジェクト参加歴やチームでの貢献，対人調整能力などの非制度的な実績の妥当性検証にも応用が可能である．実践に基づく履歴や評価を明示的な基準と共に構造化することで，従来は暗黙的かつ属人的な判断に依存していたスキルや実績の証明を，情報システム上で実装可能な形に再構成できる．これにより，リモートワーク，フリーランスといった多様な協調プロセスにおける人材評価やスキル証明のあり方にも，妥当性検証の視点を導入可能となる．

本研究の意義は，デジタル証明書の活用を初めとして，デジタル社会に必要な様々な情報の妥当性検証を実現するモデルを確立したことにある．本研究で取り上げた商取引，教育に限らず，社会のあらゆる協調プロセスの中で，人々は情報をやりとりしている．インターネットを通じた協調プロセスが前提となった現代において，クレームの妥当性検証は，社会のあらゆる場面で必要とされる営みである．本研究を礎として，社会の中の様々な場面における属人的な判断を情報システムに置き換え，検証可能な情報流通の実現が期待される．

参考文献

[1] Trusted Web 推進協議会. Trusted web ホワイトペーパー ver 3.0 概要/コンセプト編. https://www.kantei.go.jp/jp/singi/digitalmarket/trusted_web/pdf/trustedweb_concept.pdf, 2023. accessed: 2025-06-02.

[2] Trusted Web 推進協議会. Trusted web ホワイトペーパー ver 3.0 ユースケース編. https://www.kantei.go.jp/jp/singi/digitalmarket/trusted_web/pdf/trustedweb_usecase.pdf, 2023. accessed: 2025-06-02.

[3] Trusted Web 推進協議会. Trusted web ホワイトペーパー ver 3.0 実装編. https://www.kantei.go.jp/jp/singi/digitalmarket/trusted_web/pdf/trustedweb_implementation.pdf, 2023. accessed: 2025-06-02.

[4] Shaina Raza and Chen Ding. Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics*, 13:1–28, March 2022.

[5] Esma Aimeur, Sabrine Amri, and Gilles Brassard. Fake news, disinformation and misinformation in social media: a review. *Social Network Analysis and Mining*, 13, February 2023.

[6] Yang Liu and Yi-Fang Wu. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proc. the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, pages 354–361, April 2018.

[7] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3):171–188, 2020.

[8] Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5), September

2020.

[9] V Stafford. Zero trust architecture. *NIST special publication*, 800:207, 2020.

[10] Muhammad Ajmal Azad, Sidrah Abdullah, Junaid Arshad, Harjinder Lallie, and Yussuf Hassan Ahmed. Verify and trust: A multidimensional survey of zero-trust security in the age of iot. *Internet of Things*, 27:101227, 2024.

[11] Christoph Buck, Christian Olenberger, André Schweizer, Fabiane Völter, and Torsten Eymann. Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. *Computers & Security*, 110:102436, 2021.

[12] International Organization for Standardization. IT Security and Privacy A framework for identity management Part 1: Terminology and concepts. Standard, International Organization for Standardization, February 2023.

[13] International Organization for Standardization. Cards and security devices for personal identification Building blocks for identity management via mobile devices Part 1: Generic system architectures of mobile eID systems. Standard, International Organization for Standardization, February 2023.

[14] Manu Sporny, Dave Longley, and David Chadwick. Verifiable credentials data model v1.1. Recommendation, W3C, March 2022.

[15] Manu Sporny, Dave Longley, David Chadwick, and Ivan Herman. Verifiable credentials data model v2.0. Recommendation, W3C, May 2025.

[16] Abylay Satybaldy, Md. Sadek Ferdous, and Mariusz Nowostawski. A taxonomy of challenges for self-sovereign identity systems. *IEEE Access*, 12:16151–16177, 2024.

[17] Carlo Mazzocca, Abbas Acar, Selcuk Uluagac, Rebecca Montanari, Paolo Bellavista, and Mauro Conti. A survey on decentralized identifiers and verifiable credentials. `https://arxiv.org/abs/2402.02455`, 2024. accessed: 2025-06-02.

[18] デジタル庁. 新型コロナワクチン接種証明書アプリ. `https://www.digital.go.jp/policies/vaccinecert`, 2024. accessed: 2025-06-02.

[19] 1 EdTech. Open badges specification candidate final public spec version 3.0. `https://www.imsglobal.org/spec/ob/v3p0/`, May 2024. accessed: 2025-06-02.

[20] 1 EdTech. Open badges implementation guide. `https://www.imsglobal.org/spec/ob/v3p0/impl`, April 2024. accessed: 2025-06-02.

[21] Mitchell Landrigan, Stephen Wilson, and Hamish Fraser. Why are there so

many digital identities? *Law, Technology and Humans*, 6(1):1–18, 2024.

[22] Phillip Shoemaker. What Is Reusable Identity? https://www.identity.com/what-is-reusable-identity/, 2024. accessed: 2025-06-02.

[23] Mike Tuchen. A Future Of Reusable Identity Is On The Way. https://www.forbes.com/councils/forbestechcouncil/2023/11/13/a-future-of-reusable-identity-is-on-the-way/, 2024. accessed: 2025-06-02.

[24] Tobias Looker, Vasilis Kalos, Andrew Whitehead, and Mike Lodder. The bbs signature scheme. https://datatracker.ietf.org/doc/draft-irtf-cfrg-bbs-signatures/, December 2023. accessed: 2025-06-02.

[25] Daniel Fett, Kristina Yasuda, and Brian Campbell. Selective disclosure for jwts (sd-jwt). https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/, September 2024. accessed: 2025-06-02.

[26] 小山 虎. **信頼を考える**. 勁草書房, 2018.

[27] 富士榮尚寛, 鈴木茂哉, 阿部涼介, and 貞弘崇行. デジタルクレデンシャルの利用用途に応じた管理要件に関する考察. https://dal.sfc.keio.ac.jp/ja/TR/management-requirements-for-digital-credentials/, 2025. accessed: 2025-06-02.

[28] Nicolas Serrano, Hilda Hadan, and L Jean Camp. A complete study of pki (pki's known incidents). In *Proc. TPRC47: The 47th Research Conference on Communication, Information and Internet Policy*, July 2019.

[29] Hilda Hadan, Nicolas Serrano, and L Jean Camp. A holistic analysis of web-based public key infrastructure failures: comparing experts' perceptions and real-world incidents. *Journal of Cybersecurity*, 7(1):tyab025, December 2021.

[30] Skyler Johnson, Katherine Ferro, L. Jean Camp, and Hilda Hadan. Human and organizational factors in public key certificate authority failures. In *Proc. the 2021 ACM SIGSAC Conference on Computer and Communications Security*, page 2414–2416, 2021.

[31] Hakan Yildiz, Axel Küpper, Dirk Thatmann, Sebastian Göndör, and Patrick Herbke. Toward interoperable self-sovereign identities. *IEEE Access*, 11:114080–114116, 2023.

[32] Oxford University Press. Oxford English Dictionary, September 2023.

[33] International Organization for Standardization. Quality management systems — Fundamentals and vocabulary. Standard, International Organization for

Standardization, September 2015.

[34] National Institute of Standards and Technology. Engineering trustworthy secure systems. Technical Report NIST SP 800-160 Vol. 1 Rev. 1, U.S. Department of Commerce, November 2022.

[35] National Institute of Standards and Technology. Cybersecurity supply chain risk management practices for systems and organizations. Technical Report NIST SP 800-161 Rev. 1, U.S. Department of Commerce, May 2022.

[36] National Institute of Standards and Technology. Notional supply chain risk management practices for federal information systems. Technical Report NIST IR 7622, U.S. Department of Commerce, October 2012.

[37] Erika Sugita, Ryosuke Abe, Shigeya Suzuki, Keisuke Uehara, and Osamu Nakamura. A system for selective disclosure of information about a patient with intractable disease. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1482–1487, 2023.

[38] Rafael Belchior, Benedikt Putz, Guenther Pernul, Miguel Correia, André Vasconcelos, and Sérgio Guerreiro. Ssibac: Self-sovereign identity based access control. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1935–1943, 2020.

[39] Michael Kuperberg. Blockchain-based identity management: A survey from the enterprise and ecosystem perspective. *IEEE Transactions on Engineering Management*, 67(4):1008–1027, 2020.

[40] Nabil Al-Najjar, Luciano Pomatto, and Alvaro Sandroni. Claim validation. *American Economic Review*, 104(11):3725–36, November 2014.

[41] Nicholas Weber and Sebastian Karcher. Seeking justification: How expert reviewers validate empirical claims with data annotations. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, page 227–234. Association for Computing Machinery, 2020.

[42] Gustavo Lúcius Fernandes and Pedro O. S. Vaz-de Melo. Between acceptance and rejection: challenges for an automatic peer review process. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*. Association for Computing Machinery, 2022.

[43] Simon Price and Peter A. Flach. Computational support for academic peer review: a perspective from artificial intelligence. *Commun. ACM*, 60(3):70–79, February 2017.

[44] Nihar B. Shah. Challenges, experiments, and computational solutions in peer

review. *Commun. ACM*, 65(6):76–87, May 2022.

[45] Khubaib Ahmed Qureshi, Rauf Ahmed Shams Malick, and Muhammad Sabih. Social media and microblogs credibility: Identification, theory driven framework, and recommendation. *IEEE Access*, 9:137744–137781, 2021.

[46] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[47] Don Johnson and Scott Menezes, Alfredand Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security*, 1(1):36–63, August 2001.

[48] Michael B. Jones, John Bradley, and Nat Sakimura. Rfc 7515 JSON Web Signature (JWS). Technical report, Internet Engineering Task Force, May 2015.

[49] Michael B. Jones, John Bradley, and Nat Sakimura. Rfc 7519 JSON Web Token (JWT). Technical report, Internet Engineering Task Force, May 2015. accessed: 2025-06-02.

[50] Dave Longley, Manu Sporny, and Ivan Herman. Verifiable credential data integrity 1.0. Recommendation, W3C, May 2025.

[51] Securing verifiable credentials using jose and cose. Recommendation, W3C, May 2025.

[52] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`, 2008. accessed: 2025-06-02.

[53] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. `https://ethereum.github.io/yellowpaper/paper.pdf`. accessed: 2025-06-02.

[54] Shi-Yi Lin, Lei Zhang, Jing Li, Li-li Ji, and Yue Sun. A survey of application research based on blockchain smart contract. *Wireless Networks*, 28(2):635–690, February 2022.

[55] Pratima Sharma, Rajni Jindal, and Malaya Dutta Borah. A review of smart contract-based platforms, applications, and challenges. *Cluster Computing*, 26(1):395–421, February 2023.

[56] Shuai Wang, Yong Yuan, Xiao Wang, Juanjuan Li, Rui Qin, and Fei-Yue Wang. An overview of smart contract: Architecture, applications, and future trends. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 108–113, 2018.

[57] Fabian Vogelsteller and Vitalik Buterin. Eip 20: Erc-20 token standard. `https://eips.ethereum.org/EIPS/eip-20`, 2015. accessed: 2025-06-02.

[58] Maurice Herlihy. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, PODC '18, page 245–254, New York, NY, USA, 2018. Association for Computing Machinery.

[59] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, page 7–17, New York, NY, USA, 1997. Association for Computing Machinery.

[60] Iddo Bentov and Ranjit Kumaresan. How to use bitcoin to design fair protocols. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 421–439, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[61] Markus Jakobsson. Ripping coins for a fair exchange. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, pages 220–230, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[62] Jethro G. Beekman. A denial of service attack against fair computations using bitcoin deposits. *Information Processing Letters*, 116(2):144–146, 2016.

[63] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. Secure multiparty computations on bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 443–458, 2014.

[64] Steven Goldfeder, Joseph Bonneau, Rosario Gennaro, and Arvind Narayanan. Escrow protocols for cryptocurrencies: How to buy physical goods using bitcoin. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security*, pages 321–339, Cham, Switzerland, 2017. Springer International Publishing.

[65] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. Fairshare: Blockchain enabled fair, accountable and secure data sharing for industrial iot. *IEEE Transactions on Network and Service Management*, 20:2929–2941, 2023.

[66] Xuan Son Ha, Trieu Hai Le, Tan Tai Phan, Hung Huy Duc Nguyen, Hong Khanh Vo, and Nghia Duong-Trung. Scrutinizing trust and transparency in cash on delivery systems. In Guojun Wang, Bing Chen, Wei Li, Roberto Di Pietro, Xuefeng Yan, and Hao Han, editors, *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pages 214–227, Cham, Switzerland, 2021. Springer International Publishing.

[67] James Meijers, Guntur Dharma Putra, Grammateia Kotsialou, Salil S. Kanhere, and Andreas Veneris. Cost-effective blockchain-based iot data marketplaces with a credit invariant. In *Proc. 2021 IEEE International Conference*

on Blockchain and Cryptocurrency (ICBC), pages 1–9, Sydney, Australia, 2021.

[68] Umer Majeed, Latif U. Khan, Ibrar Yaqoob, S.M. Ahsan Kazmi, Khaled Salah, and Choong Seon Hong. Blockchain for iot-based smart cities: Recent advances, requirements, and future challenges. *Journal of Network and Computer Applications*, 181:103007, 2021.

[69] Stefan Dziembowski, Lisa Eckey, and Sebastian Faust. Fairswap: How to fairly exchange digital goods. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 967–984, New York, NY, USA, 2018. Association for Computing Machinery.

[70] Aditya Asgaonkar and Bhaskar Krishnamachari. Solving the buyer and seller' s dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator. In *Proc. of 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 262–267, Seoul, Korea (South), 2019.

[71] Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. A survey on blockchain: A game theoretical perspective. *IEEE Access*, 7:47615–47643, 2019.

[72] Richard O. Duda and Edward H. Shortliffe. Expert systems research. *Science*, 220(4594):261–268, 1983.

[73] Joseph C. Giarratano and Gary D. Riley. *Expert Systems: Principles and Programming*. Brooks/Cole Publishing Co., 2005.

[74] Frederick Hayes-Roth. Rule-based systems. *Commun. ACM*, 28(9):921–932, September 1985.

[75] Zhi-Jie Zhou, Guan-Yu Hu, Chang-Hua Hu, Cheng-Lin Wen, and Lei-Lei Chang. A survey of belief rule-base expert system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(8):4944–4958, 2021.

[76] Bassem S. Abu-Nasser. Medical expert systems survey. *International Journal of Engineering and Information Systems*, 1(7):218–224, 2017.

[77] Samy Abu-Naser, Al-Dahdooh R., Mushtaha A., and El-Naffar M. Knowledge management in esmda: Expert system for medical diagnostic assistance. *Journal of Artificial Intelligence and Machine Learning. Journal*, 10, January 2010.

[78] Guilan Kong, Dong-Ling Xu, Richard Body, Jian-Bo Yang, Kevin Mackway-Jones, and Simon Carley. A belief rule-based decision support system for clinical risk assessment of cardiac chest pain. *European Journal of Operational*

Research, 219(3):564–573, 2012.

[79] Jian-Bo Yang, Jun Liu, Jin Wang, How-Sing Sii, and Hong-Wei Wang. Belief rule-base inference methodology using the evidential reasoning approach-rimer. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(2):266–285, 2006.

[80] Douglas B. Lenat and Edward A. Feigenbaum. On the thresholds of knowledge. *Artificial Intelligence*, 47(1):185–250, 1991.

[81] Evandro S. Ortigossa, Thales Gonçalves, and Luis Gustavo Nonato. Explainable artificial intelligence (xai)—from theory to methods and applications. *IEEE Access*, 12:80799–80846, 2024.

[82] R.S. Michalski and R.L. Chilausky. Knowledge acquisition by encoding expert rules versus computer induction from examples: a case study involving soybean pathology. *International Journal of Man-Machine Studies*, 12(1):63–87, 1980.

[83] Marc Bezem. Consistency of rule-based expert systems. In Ewing Lusk and Ross Overbeek, editors, *Proc. 9th International Conference on Automated Deduction*, pages 151–161, 1988.

[84] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1135–1144, 2016.

[85] H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):358–366, 1953.

[86] Nat Sakimura, John Bradley, Mike Jones, Breno de Medeiros, and Chuck Mortimore. OpenID Connect Core 1.0 incorporating errata set 1. Technical report, OpenID Foundation, 2014.

[87] T. Lodderstedt, D. Fett, M. Haine, A. Pulido, K. Lehmann, and K. Koiwai. Openid connect for identity assurance 1.0. Technical report, OpenID Foundation, 2024.

[88] 犯罪による収益の移転防止に関する法律. `https://laws.e-gov.go.jp/law/419AC0000000022/`, 2007. accessed: 2025-06-02.

[89] CA/Browser Forum. Baseline requirements for the issuance and management of publicly-trusted certificates. `https://cabforum.org/baseline-requirements/`, 2023. accessed: 2025-06-02.

[90] Lisa Eckey, Sebastian Faust, and Benjamin Schlosser. Optiswap: Fast optimistic fair exchange. In *Proceedings of the 15th ACM Asia Conference on*

Computer and Communications Security, ASIA CCS '20, page 543–557, New York, NY, USA, 2020. Association for Computing Machinery.

[91] Sepideh Avizheh, Preston Haffey, and Reihaneh Safavi-Naini. Privacy-preserving fairswap: Fairness and privacy interplay. *Proceedings on Privacy Enhancing Technologies*, 2022.

[92] Simon Janin, Kaihua Qin, Akaki Mamageishvili, and Arthur Gervais. Filebounty: Fair data exchange. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 357–366, 2020.

[93] hardhat. `https://hardhat.org/`. accessed: 2025-06-02.

[94] Ankit Gangwal, Haripriya Ravali Gangavalli, and Apoorva Thirupathi. A survey of layer-two blockchain protocols. *Journal of Network and Computer Applications*, 209:103539, 2023.

[95] Ethereum improvement proposals - ercs. `https://eips.ethereum.org/erc`. accessed: 2025-06-02.

[96] William Entriken, Dieter Shirley, Jacob Evans, and Nastassia Sachs. ERC-721: Non-Fungible Token Standard. `https://eips.ethereum.org/EIPS/eip-721`, January 2018. accessed: 2025-06-02.

[97] Witek Radomski, Andrew Cooke, Philippe Castonguay, James Therien, Eric Binet, and Ronan Sandford. ERC-1155: Multi Token Standard. `https://eips.ethereum.org/EIPS/eip-1155`, June 2018. accessed: 2025-06-02.

[98] International Organization for Standardization. Information technology — Object Management Group Business Process Model and Notation. Standard, International Organization for Standardization, July 2013.

[99] Flavio Corradini, Alessandro Marcelletti, Andrea Morichetta, Andrea Polini, Barbara Re, and Francesco Tiezzi. Flexible execution of multi-party business processes on blockchain. In *Proceedings of the 5th International Workshop on Emerging Trends in Software Engineering for Blockchain*, WETSEB '22, page 25–32, New York, NY, USA, 2023. Association for Computing Machinery.

[100] Flavio Corradini, Alessandro Marcelletti, Andrea Morichetta, Andrea Polini, Barbara Re, and Francesco Tiezzi. A flexible approach to multi-party business process execution on blockchain. *Future Generation Computer Systems*, 147:219–234, 2023.

[101] Emanuele Bellini, Youssef Iraqi, and Ernesto Damiani. Blockchain-based distributed trust and reputation management systems: A survey. *IEEE Access*, 8:21127–21151, 2020.

[102] Alan Contreras and George Gollin. The real and the fake degree and diploma mills. *Change: The Magazine of Higher Learning*, 41(2):36–43, 2009.

[103] Paul Attewell and Thurston Domina. Educational imposters and fake degrees. *Research in Social Stratification and Mobility*, 29(1):57–69, 2011.

[104] Gilles Grolleau, Tarik Lakhal, and Naoufel Mzoughi. An introduction to the economics of fake degrees. *Journal of Economic Issues*, 42(3):673–693, 2008.

[105] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, pages 369–378, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[106] Camunda. Camunda modeler. `https://github.com/camunda/camunda-modeler`. accessed: 2025-06-02.

[107] Dave Longley, Dmitri Zagidulin, and Manu Sporny. The did:key method v0.7. `https://w3c-ccg.github.io/did-key-spec/`. accessed: 2025-06-02.

[108] Manu Sporny, Dave Longley, Markus Sabadello, Drummond Reed, Orie Steele, and Christopher Allen. Decentralized identifiers (dids) v1.0. Technical report, W3C, July 2022.

[109] Paul Bastian, Joosten Rieks, Rivai Zaïda, Terbu Oliver, Snorre Lothar von Gohren Edwin, Antonino Antonio, Fotiou Nikos, Curran Stephen, and Azeem Ahamed. Identifier binding: defining the core of holder binding. `https://github.com/WebOfTrustInfo/rwot11-the-hague/blob/master/final-documents/identifier-binding.pdf`, February 2023. accessed: 2025-06-02.

[110] Oliver Terbu. W3c verifiable credentials confidence method. `https://w3c-ccg.github.io/confidence-method-spec/`, October 2023. accessed: 2025-06-02.

[111] Henk Birkholz, Antoine Delignat-Lavaud, Cedric Fournet, Yogesh Deshpande, and Steve Lasker. An Architecture for Trustworthy and Transparent Digital Supply Chains. `https://datatracker.ietf.org/doc/draft-ietf-scitt-architecture/`, 2024. accessed: 2025-06-02.

[112] Ananta Soneji, Faris Bugra Kokulu, Carlos Rubio-Medrano, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, and Adam Doupé. "flawed, but like democracy we don' t have a better system": The experts' insights on the peer review process of evaluating security papers. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1845–1862, 2022.

[113] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter,

Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *Proc. 2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2023.

[114] Jing Qiu, Zhihong Tian, Chunlai Du, Qi Zuo, Shen Su, and Binxing Fang. A survey on access control in the age of internet of things. *IEEE Internet of Things Journal*, 7(6):4682–4696, 2020.

[115] 慶應義塾大学 SFC 研究所 トラステッド・インターネット・アーキテクチャー・ラボ. Trustable internet white paper v1.0. `https://tial.sfc.keio.ac.jp/blob/Trustable_Internet%E3%83%9B%E3%83%AF%E3%82%A4%E3%83%88%E3%83%9A%E3%83%BC%E3%83%91%E3%83%BCV1.0.pdf`, 2022. accessed: 2025-06-02.

[116] Xuewen Luo, Hsiao-Hwa Chen, and Qing Guo. Semantic communications: Overview, open issues, and future research directions. *IEEE Wireless Communications*, 29(1):210–219, 2022.

[117] Yating Liu, Xiaojie Wang, Zhaolong Ning, MengChu Zhou, Lei Guo, and Behrouz Jedari. A survey on semantic communications: Technologies, solutions, applications and challenges. *Digital Communications and Networks*, 10(3):528–545, 2024.

[118] Wanting Yang, Hongyang Du, Zi Qin Liew, Wei Yang Bryan Lim, Zehui Xiong, Dusit Niyato, Xuefen Chi, Xuemin Shen, and Chunyan Miao. Semantic communications for future internet: Fundamentals, applications, and challenges. *IEEE Communications Surveys & Tutorials*, 25(1):213–250, 2023.

[119] Vincent Drury and Ulrike Meyer. Certified phishing: Taking a look at public key certificates of phishing websites. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pages 211–223, Santa Clara, CA, August 2019. USENIX Association.

謝辞

本博士論文を完成させるにあたり，多くの方々のご支援とご指導をいただきました．

まず，主査である慶應義塾大学 環境情報学部植原啓介教授に深く感謝いたします．植原教授には，教育体験としての情報基礎の講義を担当させていただくきっかけを作っていただいたのみならず，今よりも遥かに未熟だった修士課程時代の自分にも期待の声をかけていただいたことが，本博士論文へ至るまでの大きな支えとなりました．

また，副査を務めていただいた環境情報学部 楠本博之教授，環境情報学部服部隆志教授，慶應義塾大学グローバルリサーチインスティチュート村井純特任教授，慶應義塾大学グローバルリサーチインスティチュート中村修特任教授， 政策・メディア研究科鈴木茂哉特任教授にも深く感謝申し上げます．楠本教授には，修士課程時代より自身では気づかない鋭い視点からのご指摘を何度もいただきました．服部教授には，情報基礎の講義を担当させていただいたのみならず，本論文の仕上げにあたり重要なご指摘をいただきました．村井特任教授には，自分にブロックチェーンを入り口として “トラスト” という分野の研究に取り組む大きなきっかけをいただきました．中村特任教授の，“研究により開かれる世界観” を大事にした御指導は，本博士論文の核となるアイディアに至るまでの数年間の中で大きな成長のきっかけとなりました．当初ブロックチェーンという 1 つの技術に興味を持っただけの自分が，抽象モデルの議論から成り立つ本博士論文を完成させるに至るには，中村教授の御指導がなければ成し得ませんでした．また，鈴木特任教授には，自身の状況にも深くご理解を示していただきながら，日々の研究活動を最も身近で支え，研究のイロハもわからない自分に辛抱強くご指導いただきました．鈴木特任教授の伴走がなければ，間違いなく今日の自分はありません．不出来な弟子で 6 年以上もかかってしまいましたが，鈴木特任教授の下で博士課程をやり抜けたことを，光栄に思います．深く，感謝いたします．

続いて，村井・中村合同研究室の教員の皆様，環境情報学部 Rodney D. Van Meter 教授，三木仁教授，中澤仁教授，大越匡准教授には，学部生時代から長きにわたりご指導いただいたことに感謝いたします．また，政策・メディア研究科佐藤雅明特任准教授，工藤紀篤特任助教，松谷健史博士，メディアデザイン研究科永山翔太准教授には，時には教員として，時には頼りになる先輩方として，公私共に様々な面でのアドバイスをいただきま

した．永山准教授が博士論文を書く姿を見て，自分もいつか成し遂げたいと思ったことが，博士課程進学の 1 つのきっかけでもありました．また，現 早稲田大学 大学院経営管理研究科斉藤賢爾教授をはじめとした NECO グループで，学部生の頃に議論させていただいたことが，研究の道へ進む最初の一歩でした．

さらに，時に深い闇へ飲まれそうになる博士課程を生き抜いた戦友である豊田安信博士，川本章太氏，Korry Luke 氏，水野史暁氏，石原匠氏に感謝いたします．分野はそれぞれ違えど，皆様と光を目指して戦い抜いた日々がなければ，決して博士論文の完成に至ることはできませんでした．今後の人生でも，お互い切磋琢磨する良き友であり続けたいと思っています．

村井・中村合同研究室の先輩・後輩の皆様に感謝いたします．特に，旧 bcali グループおよび旧 Kumo グループ時代を含めた，Delight グループメンバに深く感謝いたします．みんなと議論，談笑し，ブロックチェーン，トラスト，分散システムに関して熱く語り合ったことが，博士論文を完成させるまでの重要な過程であったと感じています．特に，博士課程前半のコロナ禍において，Verifiable Credentials および Decentralized Identifiers に関して多くの議論を共にした菅井豪留氏，杉田瑛里香氏，押川拓夢氏に感謝いたします．3 人と議論したことが，博士論文の大きな下敷きとなりました．また，望月理来氏には，博士論文の根幹を成すジャーナル論文の共著者としてのみならず，多くの議論を共にさせていただきました．さらに，竹村太希氏，坂内理人氏，曽我悠真氏，椎葉瑠星氏には，博士論文の執筆の佳境や公聴会への準備の段階で，多くの有益なコメントをいただきました．

村井純研究室秘書の渡辺康恵氏，宇佐美由美子氏に感謝いたします．出張をはじめとした様々な手続きにおいて，裏方から研究生活を支えていただきました．

また，村井・中村合同研究室の学部時代，修士課程時代の同期の皆様に感謝いたします．優秀すぎる同期たちの中で，いわばブロックチェーンという飛び道具をテーマとすることで張り合おうとしたことが，博士課程へと至るきっかけでした．社会の中でのみんなの活躍が，研究生活における良き刺激となりました．

次に，本博士論文の執筆に至るまでに共に議論させていただいた共同研究先の方々に感謝いたします．株式会社メルカリ様との共同研究では，本論文のケーススタディの一部となる議論を共にさせていただきました．富士通株式会社様との共同研究では，情報の検証可能性にまつわる広範な議論をさせていただきました．伊藤忠テクノソリューションズ株式会社様との共同研究では，信頼に関する深い議論を共にさせていただきました．それぞれの共同研究でご議論させていただいたことのエッセンスが，本博士論文へと至る重要な過程でした．

また，WIDE プロジェクトの皆様に感謝いたします．皆様と活動を共にしたことにより，幅広い視点を得たことが大きな成長とつながりました．

博士課程在学中に奨学金を給付いただいた，慶応工学会に感謝いたします．また，慶應義塾大学湘南藤沢キャンパス学生支援担当の皆様にも感謝いたします．皆様のご支援がなければ，育児と学生生活を両立させ，本論文の完成には至りませんでした．

Satoshi Nakamto を名乗る匿名の集団ないしは個人にも感謝いたします．氏の Bitcoin の発明がなければ，私が研究の道に進むこともありませんでした．今日の技術にまつわる狂乱から，氏が夢見たであろう世界への一歩を示せたのではないかと思います．また，Vitalik Buterin 氏の Ethereum とスマートコントラクトの発明にも感謝いたします．研究の中での多くのプロトタイプは，氏の発明なくしては実現できませんでした．プロトタイプを超え，社会の基盤たりうる技術へと発展，変貌していくことを願って止みません．

Philip Morris International Inc. にも感謝いたします．御社の偉大な製品である Marlboro および IQOS が，長く厳しい研究生活を生き抜く休息時間を彩りました．

次に，自分の敬愛する Slipknot，Bring Me Horizon，Crossfaith，マキシマムザ ホルモン，Bullet For My Valentine，Enter Sikari，BABYMETAL，A Ghost of Flare，Sable Hills，Earthists.， Bloodywood，Memphis May Fire，a crowd of rebellion，Survive Said The Prophet，YOASOBI，Ado，彼女 in the display，眩暈 SIREN，その他にも書ききれないほどのアーティストたちに感謝いたします．重低音は，私の研究生活の中で欠かせないものでした．また，自身が所属していたバンドである MEL，Layout my Torturechamber のメンバーをはじめとしたライブハウス関係者の皆様にも感謝いたします．自身のバンド活動の中での想いがなければ，本研究の根底にある “存在証明” というクレームの探究の旅は始まりませんでした．いつか，本博士論文の成果が敬愛するアーティストたちの存在証明へと貢献することを願って止みません．

また，博士課程を精神面から支えていただいた友人たちにも感謝いたします．細川毅騎氏との出会いがなければ，自分が村井研究室に所属することも，この分野に飛び込むこともありませんでした．長谷川優太，まりあ夫妻にも，多くの時間を自分や家族と過ごしていただき，休息の機会を何度もいただきました．

慶應義塾大学湘南藤沢キャンパスに感謝いたします．SFC の中で学際的な学びを続けられたこと，そしてこの環境が，社会と技術を橋渡しする本博士論文の成果につながりました．

自身を温かく見守り，陰から支え続けてくれた家族，祖父 小久保博義，祖母小久保基子，母 阿部洋子，兄 阿部洵介，姉 阿部蕗子，父 阿部裕一，義父山崎和司，義母 山崎敬子にも感謝いたします．長きにわたる学生生活と，並行した私の人生の進展を支え，見守っていたかなければ，本博士論文の執筆には至りませんでした．唯一，祖父に自身の博士号を見せることが叶わなかったことは，残念でなりません．

また，生活を共にした愛犬たち，ルル，ココ，ゼットンにも感謝します．君たちとの散歩の途中で思いついたアイディアが，何度研究を次に進めるきっかけになったかわかり

ません．

長女 維吹，長男 環玄にも深く，感謝いたします．家にいない日も多く，寂しい思いをさせてしまう父に，いつでも屈託のない笑顔を見せてくれてありがとう．君たちがいなければ，ここまで走り切ることはできませんでした．いつか父の取り組みが，君たちが将来生きる社会に貢献することを願って止みません．

最後に，最愛の妻 綾香に感謝いたします．想像以上に長く厳しい道であった博士論文へ至る道を，暖かく，時に厳しく鼓舞しながら，支え続けてくれました．小難しいことを考えつづけ，信念を曲げず，頑固な自分に対して，本当に辛抱強く信じてくれた君がいなければ，到底博士論文を書き切ることはできませんでした．この博士論文を，綾香に捧げることとし，謝辞の筆を置きます．

付録 A

Shinken モデルに基づくクレーム検証基準の Prolog による記述

本付録では，本論文で提案した Shinken モデルに基づくクレーム検証基準の Prolog での記述を示す．なお，本付録中では，Prolog のコードとして定義される述語を predicate_name/**引数数**の形で表現する．

A.1 署名付きデータに対する操作の表現

本論文では，クレームの検証基準を構成するため，署名付きデータの活用を取り上げて議論した．本論文においては，具体的な署名アルゴリズムに依存せず，抽象化された形で署名付きデータのモデルを整理した．そこで，本論文で議論した検証基準の Prolog による記述のために，3.1 章で議論したデジタル署名に対する操作と署名付きデータのモデルを Prolog で実装した．実装したモデルをコード A.1 に示す．

まず，署名鍵 (Signing Key) と検証鍵 (Verification Key) の関係性を表現するため，述語 pair/2 を定義する．述語 pair/2 は，第一引数に署名鍵 (SigningKey)，第二引数に検証鍵 (VerificationKey) を取り，2 つの鍵が非対称暗号における対応する鍵ペアであることを示す．また，デジタル署名の生成と検証の操作を表す述語 create_signature/3 と verify_signature/3 を定義する．述語 create_signature/3 の第一引数は署名鍵，第二引数は署名対象のデータ (Payload) であり，第三引数として生成される署名 (Signature) を取る．本実装では，特定の署名アルゴリズムに依存しない形でのデジタル署名の操作のエミュレーションとして，署名の実態は署名鍵と署名対象のデータを結合した値とした．述語 verify_signature/3 は第一引数に検証鍵，第二引数に署名対象のデータ，第三引数に署名を取る．述語 verify_signature/3 は第一引数の検証鍵に対応した署名鍵を用いて，署名が create_signature/3 で生成された値であれば真となる．以上を用いて，デジタル署名に対する操作を Prolog で表現できる．

また，3.1.2 節で示した署名付きデータを示す構造項 signed_data/3 を定義する．構造項 signed_data/3 は，署名鍵 (VerificationKey)，対象データ (Payload)，署名 (Signature) から構成される．述語 pair/2 と構造項 signed_data/3 を用いて，述語 create_signed_data/3 および verify_signed_data/1 をそれぞれ定義する．述語 create_signed_data/3 の引数は，署名鍵，対象データ，署名が付与された署名付きデータをそれぞれ示す．述語 create_signed_data/3 は，以下の全ての条件を満たす時，真となる．

- 第一引数である署名鍵に対応する検証鍵が，第三引数の署名付きデータに含まれる検証鍵である
- 第二引数である対象データが，第三引数の署名付きデータに含まれる対象データである
- 第三引数に含まれる署名が，第一引数の署名鍵および第二引数の対象データを用いて create_signature/3 で生成された値である

次に，署名付きデータの署名検証をする述語である verify_signed_data/1 を定義する．述語 verify_signed_data/1 は，署名付きデータを引数とする．当該署名付きデータが，自身に含む検証鍵と対応した署名鍵およびペイロードを元に述語 create_signed_data/3 で生成されたものであるとき，verify_signed_data/1 は真である．

鍵ペアの存在を述語 pair/2 を用いて事前に定義することで，コード A.1 で署名付きデータの生成と検証を表現できる．本付録の本節以降では，コード A.1 は事前に定義されているものと仮定する．

コード A.1　署名付きデータの生成と検証の論理

```
1  % Primitive Signature Operations
2  create_signature(
3    SigningKey,
4    Payload,
5    Signature
6    ):-
7    pair(SigningKey, VerificationKey),
8    % The following sentence is a dummy implementation of signing
         operation.
9    nonvar(Payload),
10   term_to_atom(Payload, AtomPayload),
11   atomic_list_concat([SigningKey, AtomPayload], "-", Signature).
12
13 verify_signature(VerificationKey, Payload, Signature):-
14   pair(SigningKey, VerificationKey),
15   create_signature(
16     SigningKey,
```

```
17      Payload,
18      Signature).
19
20  % Operations for Signed Data
21  create_signed_data(SigningKey, Payload, signed_data(VerificationKey,
        Payload, Signature)):-
22    pair(SigningKey, VerificationKey),
23    create_signature(SigningKey, Payload, Signature).
24
25  verify_signed_data(signed_data(VerificationKey, Payload, Signature)):-
26    pair(SigningKey, VerificationKey),
27    create_signed_data(SigningKey, Payload, signed_data(VerificationKey,
          Payload, Signature)).
```

A.2 5.4.2 節における検証基準

5.4.2 節では，署名付きデータを用いてクレーム検証を実現するための検証基準の構成を議論した．コード A.2 に，構成した検証基準の Prolog による記述を示す．

5.4.2 節の議論では，鍵とエンティティの紐付きは前提として議論した．したがって，述語 `pair/2` を用いて署名者の用いる鍵ペアの存在と，述語 `binded/2` によって，変数である署名者 (`signer`) と，検証鍵 (`verification_key_signer`) は紐づくことを示した (3 行目より 5 行目)．また，5.4.2 節では，Shinken モデルに基づき，“特定の命題 (署名の意図) が真である時に署名者は署名を付与すること” を信頼し，仮定する．したがって，述語 `sign_according_to/2` を定義し，第一引数のエンティティが，第二引数の命題が真である場合，署名を付与することを表現した (7 行目)．以上のそれぞれの述語を用いて，述語 `validated/2` を定義した (17 行目より 20 行目)．述語 `validated/2` は，第一引数に検証対象のクレームである署名の意図，第二引数に署名付きデータを取る．また，以下の 3 つの述語が全て真の時，述語 `validated/2` は真となる．

- 署名者と署名付きデータに含まれる検証鍵が紐づいている
- 署名者が署名の意図が真である場合に署名を付与する
- 署名付きデータが検証できる

コード A.2 では，述語 `sign_according_to/2` を fact として記載することで，“署名の意図が真である時に署名者が署名を付与すること” を信頼し，真と仮定することを表現した．当該箇所を信頼しない場合，以下の述語から構成される述語として表現できる．

- 署名の意図である命題が真である
- 署名者と検証鍵が紐づいている
- 当該検証鍵に対応する署名鍵によって署名付きデータが作成されている

一方，“署名の意図である命題が真である”ことは，必ずしも計算機で決定可能な問題に落とし込めるとは限らない．したがって，Shinken モデルに基づき “署名の意図が真である時に署名者が署名を付与すること”を仮定することで，決定性のある検証基準を構成できる．

コード A.2　署名付きデータを用いた署名の意図の検証

```
1  %  === Validation Logic ===
2  % Trusted Predicates
3  pair(signing_key_signer, verification_key_signer).
4  % NOTE: Binding between a key and an entity is out of scope in this
       section.
5  binded(signer, verification_key_signer).
6
7  sign_according_to(signer, intent).
8  % NOTE: If the sentence is not trusted, following logic should be
       validated.
9  % But following logic is not decidable, especially is_true(Intent).
10 % sign_according_to(Signer, Intent):-
11 %   is_true(Intent),
12 %   binded(Signer, VerificationKey),
13 %   pair(SigningKey, VerificationKey),
14 %   create_signed_data(SigningKey, _, _).
15
16 %% Logic Body
17 validated(Intent, signed_data(VerificationKey, Payload, Signature)):-
18   binded(Signer, VerificationKey),
19   sign_according_to(Signer, Intent),
20   verify_signed_data(signed_data(VerificationKey, Payload, Signature)).
```

A.3　5.4.3 節における検証基準

5.4.3 節では，5.4.2 節における検証基準を基に，デジタル証明書を用いたクレームの検証基準とその更改に関して議論した．本節では，更改前後の検証基準の Prolog による実装について述べる．

A.3.1　更改前の検証基準

前節で議論したコード A.2 を基に，デジタル証明書を用いたクレームの検証基準をコード A.3 に示す．前節の議論と同様に，更改前の検証基準では鍵とエンティティの紐付きは前提とする (3 行目から 4 行目)．デジタル証明書の場合，検証者は “認定者は対象のクレームが検証できれば証明書を発行すること”を信頼し，仮定する．, `sign_according_to/2` の第二引数として，述語 `validated/2` によって任意のクレームを，“任意の根拠情報を基に検証済みであること”が真である時，署名を付与することを

表現した (5 行目).

検証者がクレーム検証する述語 validated/2 は，クレーム自体と，デジタル証明書である署名付きデータを引数とする (8 行目)．前節で議論した検証基準の中で，署名の意図として証明書に含まれるクレームが検証済みである時，述語 validated/2 は真となる．

コード A.3　証明書を用いたクレームの検証

```
1  %  === Validation Logic ===
2  % Trusted Predicates
3  pair(signing_key_certifier, verification_key_certifier).
4  binded(certifier, verification_key_certifier).
5  sign_according_to(certifier, validated(Claim, _)).
6
7  %% Logic Body
8  validated(Claim, signed_data(VerificationKey, Claim, Signature)):-
9    binded(Signer, VerificationKey),
10   sign_according_to(Signer, validated(Claim, _)),
11   verify_signed_data(signed_data(VerificationKey, Claim, Signature)).
```

A.3.2　更改後の検証基準

コード A.3 では，検証鍵と認定者の紐付きを信頼し，仮定することで，“意図に沿って証明書を発行する認定者が，署名を付与した証明書である” ことを演繹的に推論し，検証基準を構成した．5.4.3 節では，当該仮定が破られた場合を想定し，公開鍵証明書を用いてその紐付きを確認するよう検証基準を更改した．更改後の検証基準の Prolog による実装をコード A.4 に示す．

更改後の検証基準では，直接認定者と鍵の紐付きを仮定せずに，公開鍵証明書が鍵の紐付きを示していることを確認する．，コード A.3 で仮定されていた binded(certifier, verification_key_certifier) を公開鍵証明書を用いた紐付きの確認に置き換えることで，検証基準を更改した．一方，公開鍵証明書もデジタル証明書の一種であり，署名付きデータである．，署名の意図として “対象のエンティティと検証鍵の紐付き” が真である時に公開鍵証明書の発行者が当該公開鍵証明書を発行することを信頼し，仮定することで紐付きの検証基準を構成する．A.4 では，sign_according_to(pubkey_cert_issuer, binded(Entity, VerificationKey)) によって，当該仮定を表現した (9 行目)．また，述語 validated/3 をクレーム，検証対象のクレームを含む証明書に加え，公開鍵証明書を引数とするように拡張した．これによって，公開鍵証明書を “認定者と検証鍵の紐付き” というクレームに対する証明書として再帰的に利用する検証基準を構成した (14 行目 ~23 行目).

この時，公開鍵証明書の発行者とその検証鍵の紐付きに対しても，クレームを示す証明書の検証基準と同様に仮定できない場合が考えられる．これは，物理空間中のエンティ

ティとサイバー空間中の検証鍵の紐付きは，計算機で決定困難な問題の 1 つであることから，信頼の導入によってクレームの検証を成立させる一例であると整理できる．

コード A.4　公開鍵証明書を用いた鍵と認定者の紐付きの確認を含む，証明書を用いたクレームの検証

```
1  %  === Validation Logic ===
2  % Trusted Predicates
3  pair(signing_key_certifier, verification_key_certifier).
4  pair(signing_key_pubkey_cert_issuer, verification_key_pubkey_cert_issuer
       ).
5  % binded(certifier, verification_key_certifier).
6  binded(pubkey_cert_issuer, verification_key_pubkey_cert_issuer).
7
8  sign_according_to(certifier, validated(Claim, _)).
9  sign_according_to(pubkey_cert_issuer, binded(Entity, VerificationKey)).
10
11 %% Logic Body
12 %%% Fixed logic
13 %%% validated(Claim, Certificate, PublicKeyCert)
14 validated(Claim, signed_data(VerificationKey, Claim, Signature),
       PubKeyCert):-
15   validated(binded(Certifier, VerificationKey), PubKeyCert),
16   sign_according_to(Certifier, validated(Claim, _)),
17   verify_signed_data(signed_data(VerificationKey, Claim, Signature)).
18
19 %%% Original logic: It is utilized for validation of binding between a
       key and an entity.
20 validated(Intent, signed_data(VerificationKey, Payload, Signature)):-
21   binded(Signer, VerificationKey),
22   sign_according_to(Signer, Intent),
23   verify_signed_data(signed_data(VerificationKey, Payload, Signature)).
```

A.4 6 章における検証基準

6 章では，“主張者が関与した過去の取引が正常に終了したこと” をクレームとして，検証基準を構成した．構成した検証基準の Prolog による実装をコード A.5 に示す．

まず，取引は複数タスクで構成されるプロセスであると整理し，正常終了とは全てのタスクが完了した状態であると整理した.，取引はタスクの集合として定義され，定義中のタスクが全て完了した状態であることで，特定の取引が正常に終了したとみなせる．この構造を示すために，取引の定義をその識別子とタスク定義のリストを含む構造項 `transaction_definition/2` として定義した (2 行目)．また，タスクの定義を，タスクの識別子と販売者あるいは購入者のどちらかが実行するべきものであるかを含む構造項 `task_definition/2` として定義した (3 行目)．これによって，具体的な取引の定義を記述可能である (8 行目から 15 行目)．

次に，個別の取引の履歴として，以下を引数とする構造項 transaction_history/5 を定義した.

- 取引の識別子
- 取引の定義の識別子
- 販売者の識別子
- 購入者の識別子
- タスク実行履歴のリスト

また，タスクの実行履歴として，以下を引数とする構造項 task_history/2 を定義した.

- 取引の識別子
- タスクの識別子
- タスクの実行状態

上記の履歴の構造によって，個別の取引の履歴を述語として定義可能である (17 行目から 68 行目).

取引の定義と履歴によって特定の取引の結果を検証するにあたり，履歴が改竄されていた場合, その結果が妥当とみなせるかは疑わしい．本ケーススタディでは, ブロックチェーン上で取引を実施し，その履歴がブロックチェーン上に記録されていることから，履歴が改竄されていないことを仮定した.，特定の取引の履歴がブロックチェーン上に記録されているかどうかを, 取引履歴の識別子を変数とする述語 is_recorded_on_blockchain/1 として表現した (70 行目から 72 行目)．ここで, 特定の取引が改竄されていないことを, ブロックチェーン上に記録されてることを以て仮定するために，述語 is_not_tampered/1 を定義した (77 行目).

販売者あるいは購入者として関与した特定の取引が正常終了であったことを述語 complete_tx_as_buyer/1 あるいは complete_tx_as_seller/1 として定義し，それぞれ該当する取引履歴の識別子およびエンティティの識別子を引数とする．主張者は，それぞれの述語と特定の履歴の識別子と自身の識別子を用いて，クレームを表現できる．また，検証者は，述語 validated/1 に対して，complete_tx_as_buyer/1 あるいは complete_tx_as_seller/1 を与えることで，一貫した構文で取引結果の検証を記述できるよう設計した．検証者は，与えられたクレームに対して，以下が全て真であることを検証基準として定める (81 行目から 87 行目).

- 指定された取引の識別子に対応する履歴が存在すること
- 履歴で示される購入者あるいは販売者が主張者であること

- 指定された取引が正常終了していること

中でも，“取引が正常終了していること” は，以下の検証基準で検証する (89 行目から 93 行目).

- 取引履歴が改竄されていないこと (91 行目)
- 履歴に示される取引の定義が存在すること (92 行目)
- 定義に含まれる全てのタスクに対してタスクの実行履歴が存在し，それぞれの状態が完了であること (93 行目)

以上によって，本ケーススタディにおける取引結果の検証基準を，Shinken モデルに基づき明示的な信頼と共に決定的な論理で構成した．本節で示した検証基準では，“ブロックチェーン上に記録されていることを以て，履歴が改竄されていないこと” を信頼し，仮定した．したがって，ブロックチェーン自体に対する攻撃などによって本仮定が破られた場合，取引結果を誤認する可能性がある．仮にそうした可能性を考慮する場合，“履歴が改竄されていないこと” の検証基準として，依存するブロックチェーンが攻撃を受けていないことを確認する検証基準を追加し，更改することが検討できる．

コード A.5　特定エンティティの関与した取引が正常終了であったことの検証基準

```
1  %  === Structure ===
2  % transaction_definition(DefId, [task_definition...]).
3  % task_definition(TaskId, (seller|buyer)).
4  % task_history(TxId, TaskId, Status).
5  % transaction_history(TxId, DefId, [task_history...]).
6
7  %  === Definitions ===
8  transaction_definition(
9    def1,
10   [
11     task_definition(task1, buyer),
12     task_definition(task2, seller),
13     task_definition(task3, buyer)
14   ]
15 ).
16
17 % === Histories ===
18 % Not completed
19 transaction_history(
20   tx1,
21   def1,
22   seller(alice),
23   buyer(bob),
24   [
25     task_history(task1, completed),
26     task_history(task2, completed),
```

```
    task_history(task3, completed)
  ]
).

% Not completed
transaction_history(
  tx2,
  def1,
  seller(alice),
  buyer(bob),
  [
    task_history(task1, completed),
    task_history(task2, completed),
    task_history(task3, timeouted)
  ]
).

% alice-carol transaction
transaction_history(
  tx3,
  def1,
  seller(alice),
  buyer(carol),
  [
    task_history(task1, completed),
    task_history(task2, completed),
    task_history(task3, completed)
  ]
).

% Not Recorded on Blockchain
transaction_history(
  tx4,
  def1,
  seller(alice),
  buyer(bob),
  [
    task_history(task1, completed),
    task_history(task2, completed),
    task_history(task3, completed)
  ]
).

is_recorded_on_blockchain(tx1).
is_recorded_on_blockchain(tx2).
is_recorded_on_blockchain(tx3).


%  === Validation Logics  ===
%% Trusted Predicate
```

```
77  is_not_tampered(TxId):-
78    is_recorded_on_blockchain(TxId).
79
80  %% Logics
81  validated(complete_tx_as_buyer(TxId, Buyer)):-
82    transaction_history(TxId, _, _, buyer(Buyer), _),
83    tx_completed(TxId).
84
85  validated(complete_tx_as_seller(TxId, Seller)):-
86    transaction_history(TxId, _, _, seller(Seller), _),
87    tx_completed(TxId).
88
89  tx_completed(TxId):-
90    transaction_history(TxId, DefId, _, _, TaskHistories),
91    is_not_tampered(TxId),
92    transaction_definition(DefId, TaskList),
93    all_tasks_completed(TaskList, TaskHistories).
94
95  all_tasks_completed([], _).
96  all_tasks_completed([task_definition(TaskId, _) | Rest], TaskHistories)
        :-
97    task_completed(TaskId, TaskHistories),
98    all_tasks_completed(Rest, TaskHistories).
99
100 task_completed(TaskId, [task_history(TaskId, completed) | _]).
101 task_completed(TaskId, [_ | Rest]) :-
102   task_completed(TaskId, Rest).
```

A.5 7 章における検証基準

7 章では，デジタル証明書を用いたクレーム検証における検証基準の更改と，それに付随する根拠情報のデータモデルの設計を議論した．本付録のコード A.3 で示したように，デジタル証明書を用いる場合，“証明書発行以前に認定者が対象のクレームを検証したならば，証明書を発行する” ことを信頼し，仮定することで検証基準を構成できる．一方，発行以前の認定者による検証プロセスがヒューマンエラーなどで危殆化した場合，当該仮定が破られることとなる．つまり，コード A.3 で示した検証基準のみでは，認定者によって検証されていないクレームに対する証明書が発行される可能性を否定できない．そこで，7 章では，“対象のクレームを認定者が特定の根拠情報を基に検証した上で証明書を発行したこと” を検証基準に含めるよう更改することを議論した．構成した検証基準の Prolog による記述をコード A.6 に示す．

更改前の検証基準では，述語 `sign_according_to(certifier, validated(Claim, _))` . を仮定することにより，当該認定者が証明書を発行した場合，証明書に示されるクレームは検証済みであることを仮定した．当該仮定が破られたケースに対応するため，

まず，証明書のペイロードを validated(Claim, Evidence) とすることで，“特定のクレームを，特定の根拠情報に基づき検証した” ことを表現するものとした (8 行目)．また，“特定の主体が根拠情報を確認したこと” を述語 confirmed_by/2 として定義した (14 行目から 16 行目)．述語 confirmed_by/2 は，第一引数を確認の主体，第二引数を確認対象の根拠情報とする．また，第一引数の主体に紐づく検証鍵で検証可能な署名付きデータが存在する時，述語 confirmed_by/2 は真となる．これを用いて，述語 validated/2 に，“証明書に含まれる根拠情報を認定者が確認したこと” を追加した (12 行目)．

この更改によって，証明書で示される根拠情報を確認した上で，証明書を発行したことが検証できる．したがって，もし証明書の示すクレームが妥当ではないことが明らかになった場合，証明書で示される根拠情報と認定者のポリシを照合することで，認定者の検証プロセスが危殆化していることを検知できる．

コード A.6　認定者による検証の確認を含む，証明書を用いたクレーム検証

```
1  %  === Validation Logic ===
2  % Trusted Predicates
3  binded(certifier, verification_key_certifier).
4
5  sign_according_to(certifier, validated(Claim, Evidence)).
6
7  %% Logic body
8  validated(Claim, signed_data(VerificationKey, validated(Claim, Evidence)
       , Signature)):-
9    binded(Certifier, VerificationKey),
10   sign_according_to(certifier, validated(Claim, Evidence)),
11   verify_signed_data(signed_data(VerificationKey, validated(Claim,
         Evidence), Signature)),
12   confirmed_by(Certifier, Evidence).
13
14 confirmed_by(Entity, Evidence):-
15   binded(Entity, VerificationKey),
16   verify_signed_data(signed_data(VerificationKey, validated(_, Evidence)
         , Signature)).
```

本研究に関連する発表済み成果

査読付き論文誌

本論文の一部の記述・図表は，情報処理学会および Institute of Electrical and Electronics Engineers の許諾に基づき，下記の同一著者の論文を加筆・再編集したものである．

1. “Shinken: 更改可能な透明性のあるクレーム検証モデル”
   阿部涼介，望月理来，鈴木茂哉，中村修，情報処理学会論文誌，DOI: 10.20729/0002002605, 2025-06-15 出版
2. “A Conceptual Model for Claim Validation Based on Signed Data”
   Ryosuke Abe, Shigeya Suzuki, Osamu Nakamura, IEEE Access, DOI: 10.1109/ACCESS.2024.3524509, 2024-12-31 出版

その他発表済みの成果

査読付き国際会議発表

1. “Mitigation of Seller and Buyer’s Dilemma with Transaction History and Escrow”
   Ryosuke Abe, Seiyo Kurita, Mariko Kobayashi, Shigeya Suzuki, Asian Internet Engineering Conference 2023, Hanoi, Vietnam (2023-12)
2. “A System for Selective Disclosure of Information about a Patient with Intractable Disease”
   Erika Sugita, **Ryosuke Abe**, Shigeya Suzuki, Keisuke Uehara, Osamu Nakamura, The 18th IEEE International Workshop on e-Health Systems & Web Technologies in COMPSAC 2023, Torino, Italy (2023-06)
3. “Fabchain: Managing Audit-able 3D Print Job over Blockchain,”
   Ryosuke Abe, Shigeya Suzuki, Kenji Saito, Hiroya Tanaka, Osamu Nakamura, Jun Murai, 2022 IEEE International Conference on Blockchain and

Cryptocurrency, Shanghai, China, (2022-05)

4. “Mitigating Bitcoin Node Storage Size By DHT,”
   Ryosuke Abe, Shigeya Suzuki, Jun Murai, Asian Internet Engineering Conference 2018, Bangkok, Thailand (2018-11)
5. “Attack Incentive and Security of Exchanging Tokens on Proof-of-Work Blockchain,”
   Ryosuke Abe, Keita Nakamura, Kentaro Teramoto, Misato Takahashi, Asian Internet Engineering Conference 2018, Bangkok, Thailand (2018-11)
6. “Storage Protocol for Securing Blockchain Transparency,”
   Ryosuke Abe, Hiroki Watanabe, Shigenori Ohashi, Shigeru Fujimura, Atsushi Nakadaira, The 1st IEEE International Workshop on Secure Digital Identity Management in COMPSAC 2018, Tokyo, Japan (2018-07)

国内発表

1. “Ethereum に基づいたアプリケーションの実行時間定式化の検討と計測,”
   阿部涼介, 鈴木茂哉, 研究報告マルチメディア通信と分散処理（DPS), 2020-DPS-185, (2020-12)
2. “パーソナルファブリケーション時代における Blockchain を用いた製造情報保存システム,”
   阿部涼介, 斉藤賢爾, 村井純, マルチメディア，分散協調とモバイルシンポジウム 2017 論文集, (2017-06)