

# ブロックチェーン利活用による 下請配送業者への対価支払迅速化システムの開発

下村真帆<sup>†1</sup> 古家直樹<sup>†1</sup> 小坂忠義<sup>†1</sup> 渡邊高志<sup>†1</sup>

**概要**：現状、中小企業の資金調達は課題であり、物流業界においても、長期の支払サイトが原因で、多くの中小企業の配送業者は運用資金が不足している実態がある。ところで、近年、非中央集権、耐改ざん性などの新たな価値を創出するブロックチェーンが注目されている。本研究では、ブロックチェーンを活用して、配送を他社に委託する物流業者に代わって、金融機関が配送完了後に下請配送業者に対し輸配送代金を都度払い可能なシステムを開発した。具体的には、ドライバのモバイル端末から取得した受領書画像のハッシュ値をブロックチェーンに書き込むと、下請配送業者への都度払い指示を電子債権の発行システムに通知する対価支払迅速化システムを開発した。実証実験にて、納品完了後、一次下請け、二次下請けの配送業者に対し、電子債権による都度払いができることを確認した。これにより、発注元物流業者のキャッシュアウトを変更せず、配送作業完了から対価受け取りまでの支払サイトを数週間から1日に短縮できる見込みを得た。

**キーワード**：物流、ブロックチェーン、決済

## Development of Immediate Payment System to Subcontractors by Using Blockchain

MAHO SHIMOMURA<sup>†1</sup> NAOKI FURUYA<sup>†1</sup>  
TADAYOSHI KOSAKA<sup>†1</sup> TAKASHI WATANABE<sup>†1</sup>

**Abstract**: Funding for small and medium enterprises (SMEs) in supply chain is an issue today, and in the logistics industry, due to long term payment period, many SMEs are short of running costs. On the other hand, in recent years, blockchain to create a new value such as non-centralized and tamper resistance is attracting attention. In this study, we develop a system that a financial institution can pay delivery fee each time to subcontractors on behalf of logistics company after delivery is completed by using the blockchain. Specifically, we develop an immediate payment system that notifies a digital bond issuing system of payment subcontractors after the hash value of a digital image of a signed delivery note obtained from the driver is written to the blockchain. Through PoC, we confirmed that it is possible to pay primary subcontractor and second subcontractor each time after delivery is completed. We got the prospect that payment period of subcontractors can be shortened from up to a few weeks to 1 day.

**Keywords**: logistics, blockchain, payment

### 1. はじめに

近年、物流業界ではドライバ不足や労働環境の改善、煩雑な帳票管理の解決をめざす動きが加速し、アナログ中心の見積・受発注管理、配車・運行管理業務、請求管理などをデジタル化する機運が高まっている。また、作業完了から対価受け取りまでの支払サイトが長期であることが原因で、配送業者自身が人件費や燃料費、交通費などの費用を立て替える必要があり、多くの零細企業の配送業者は運用資金が不足している。下請構造の多層化に伴い、事業者間における支払の遅延なども顕在化している実態がある [1]。

ところで、近年、非中央集権、耐改ざん性、透明性といった特徴により、新たな価値の創出を可能とするブロックチェーン (blockchain, 以下BCと略す) が注目されている。従来は中央管理者を経由する取引の形態が一般的であったが、中央管理者を挟まずユーザ同士で管理しあう取引

の形態が注目されており、手数料の削減や支払の迅速化といった価値が期待されている。BCは仮想通貨 (暗号資産) を実現する基盤技術として注目されたこともあり、海外送金や決済などの金融分野での活用が注目されている。一方で、近年、物流分野におけるBCの活用も注目されており、貿易業務の効率化や医薬品のトレーサビリティ情報管理にBCを活用するなどのユースケースも有力視されている。 [2]

本研究では、零細企業の配送業者の資金繰り改善のため、金融機関が物流業者の債務引受をすることで、物流業者が配送を委託する下請配送業者が輸配送代金を早期に受け取れるよう、配送完了後に輸配送協力会社への都度払い指示を電子債権発行システムに通知する対価支払迅速化システムを開発する。ここでは、発注情報や配送完了のエビデンスデータを改ざん不能な形で管理するためにBCを活用する。そして、対価支払迅速化システムの物流業務への適合

<sup>†1</sup> (株)日立製作所  
Hitachi Ltd.

性と BC に書き込むデータサイズの削減、機微情報の秘匿性などを検証する。

## 2. 都度払いサービスの概要とシステム要件

### 2.1 現状の受発注から支払までの業務フロー

図 1 に、物流業者の配送依頼から、輸配送、下請配送業者への支払までの現状業務の一例を示す。ここでは、物流業者が配送を委託する下請配送業者を協力会社 1、協力会社 1 が配送を委託する下請配送業者を協力会社 2 と呼ぶ。ここでは、物流業者が協力会社 1 に配送を委託後、協力会社 1 が協力会社 2 に配送を委託する場合を想定する。

まずは、発注業務について説明する。物流業者が荷主から荷物の配送依頼を受託すると、自社の車両で配送しない場合、協力会社 1 に配送を依頼する。また、協力会社 1 が配送依頼を受託後、自社の車両で配送しない場合、協力会社 2 に配送を依頼する。

つぎに、輸配送業務について説明する。ドライバーは配送日に物流業者の倉庫で荷積みをした後、物流業者の輸配送営業所の担当者から受領書を受け取り、出荷先に荷物を配送する。受領書とは、出荷先が荷主に対し荷物を受領したことの証拠として渡す文書である。出荷先の担当者が検品をし、荷物が納品されたことを確認後、ドライバーから受け取った受領書に押印する。ドライバーは納品が完了した旨を、直属の上長に連絡し、自社が所属する協力会社 2 の事務所に帰営後、受領書を上長に渡す。その後、協力会社 2 から協力会社 1、物流業者、荷主へと受領書が送付される。

最後に、輸配送代金の支払業務について説明する。支払サイトが月末締め翌月末払いの場合、物流業者から協力会社 1 への計上・支払業務は月末に実施される。物流業者は当月の締め日までに行われた配送の検収を行い、支払明細書を協力会社 1 に送付する。協力会社 1 は支払明細書の金額が間違っていないことを確認後、物流業者に対して請求書を送付する。請求書の内容に不備がなければ、翌月末に物流業者から協力会社 1 への支払が行われる。

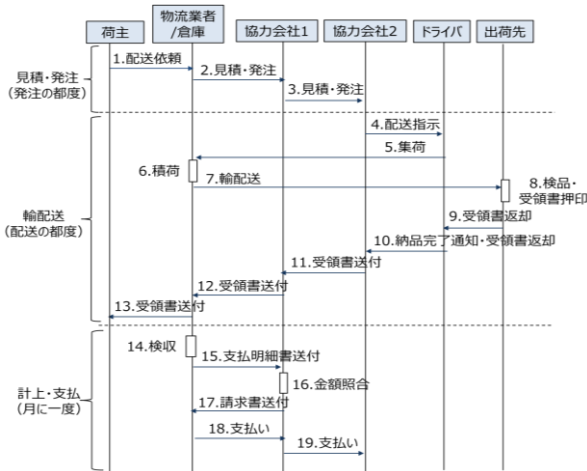


図 1 業務フロー(As Is)  
Fig. 1 Business flow (As Is)

### 2.2 下請配送業者への都度払いサービスの概要

本研究で実現をめざしている都度払いサービスの一例について説明する。月締め翌月末払いにおける作業完了から対価受け取りまでの支払サイトと都度払いサービス利用時の支払サイトを比較しながら、図 2 を用いて本サービスの概要を説明する。

支払サイトが月締め翌月末払いの場合、作業完了から下請配送業者が輸配送代金を受け取るまでに数週間かかる。一方、この都度払いサービスでは、配送完了後に、金融機関が債務引き受けを行うことで、下請配送業者が輸配送代金を早期受け取り可能な仕組みを実現する。具体的には、物流業者が配送業務の検収を行うと同時に、金融機関が輸配送代金分の電子債権を発行し、電子債権を用いて下請配送業者への債務引き受けを行う。電子債権とは、デジタルデータで表現される債権のことを言い、電子債権を用いることで都度払いが可能となる。その後、下請配送業者が任意のタイミングで債権を現金に換金することで、作業完了から下請配送業者が輸配送代金を受け取るまでの期間を数週間短縮できる。また、物流業者は従来と同じ支払サイトで金融機関への債務引受代金の決済を行うため、物流業者のキャッシュアウトは従来と変わらない。

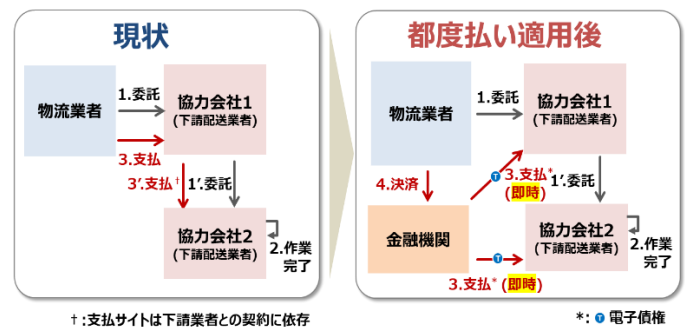


図 2 支払サイトの比較

Fig. 2 Comparison of payment terms

### 2.3 システム要件

都度払いサービスの実現をめざし、本研究では下請配送業者向け対価支払迅速化システムを開発する。主なシステムの要件としては、以下があげられる。

#### (1) 電子債権発行システムとの連携

本研究では、以下の二つの理由から、その特性を有するものの一つとして、金融機関が発行する電子債権を用いることにする。一つ目は、レートが乱高下する通貨では企業間の支払には適さないため、比較的レートが安定している、金融機関が発行する電子債権を用いて支払いを行うため。二つ目に、物流業者のキャッシュアウト時期を従来のまま維持することで、物流業者が行っている現状の支払業務への影響を少なくするためである。金融機関が債務引き受けをすることで、物流業者のキャッシュアウト不変のまま、下請配送業者のキャッシュフローを改善できる。以上の理由から、金融機関の電子債権を発行する電子債権発行シ

テムと対価支払迅速化システムを連携する必要がある。

## (2) 発注情報の管理

電子債権発行システムに電子債権の発行や移転の指示を出すために、対価支払迅速化システムは耐改ざん性を維持しつつ、電子債権の発行金額や移転金額を保持しておく必要がある。また、発注に紐づく全ての配送が完了した後、下請配送業者に発注分の輸配送代金の都度払いを行うためには、対価支払迅速化システムが配送情報と発注金額が紐づいた発注情報を保持している必要がある。

## (3) 信頼のある配送実績の取得

発注に紐づく全配送が完了した後、下請配送業者に発注金額分の都度払いをするためには、荷物の配送が完了したことを証明する情報、すなわち配送実績を対価支払迅速化システムが電子データとして取得することが求められる。

## 2.4 BC の活用と課題

都度払いサービスを、複数の物流業者や下請配送業者が利用する場合、対価支払迅速化システムは各企業が取引する発注情報や配送実績を管理する必要がある。また、誤った発注金額で都度払いしてしまうことを防ぐために、信頼のある正しい発注情報を保持する必要がある。万が一悪意を持った企業が参加しており、DB にアクセスし発注金額を改ざんして、水増し請求や歩引きなどができてしまわないよう、一度承認された発注情報は改ざんできないようにする仕組みが求められる。また、配送実績も都度払い処理のトリガーとなる重要な情報であり、信頼性の高い配送実績を取得し、管理することが求められる。信頼のある第三者機関がこれらのデータを管理することで、データの信頼性を担保する方法もあるが、管理者に悪意があればデータの中身を改ざんすることができてしまう。

そこで、第三者機関の管理を要せずに、各企業の中立性を維持しつつ、データの信頼性を保証する仕組みを構築することが求められる。それを解決する一つ的手段としてBCがあり、BCは特定の組織や企業が管理する形を取らない「非中央集権性」の分散型ネットワークを構築し、データの「透明性」、「耐改ざん性」を保証する技術として注目されている。本研究ではBCを活用し、発注情報や配送実績の信頼性を保証する。

一方で、データの透明性を保証するBCを用いてデータを管理する場合、以下の二点が課題となる。

### (1) 受発注取引に関与しない企業へのデータの秘匿化

### (2) BCを構成するノードのストレージ容量の限界

(1)について、各企業が対価支払迅速化システムの発注情報を閲覧する時、直接取引に関係ない企業は閲覧できないよう、データの秘匿性が求められる。特に、発注金額などの機微情報は、多層構造の受発注取引の場合、自社の中抜き金額を他社に知られたくないため、特に秘匿化が求めら

れる。例えば、協力会社1から協力会社2への発注金額を物流業者に秘匿化することなどが求められる。

つぎに(2)について、BCでは、全ノードが同じデータを持つことにより、高い耐障害性と耐改ざん性を実現している。しかし、対価支払迅速化システムを利用する企業が増加するほど、発注情報や配送実績などのデータをBCに書き込むトランザクション数が増えるため、BCの各ノードに求められるデータストレージ容量が増加する。国土交通省の調べによると、2020年度の宅配便取扱実績は48億3674万個であり[3]、膨大な量の発注情報や配送実績がBCに書き込まれることが予想される。各企業がノードを自社で管理する場合、ストレージの小さいノードは容量がいっぱいになり、BCの書き込みができなくなってしまうという課題がある。

本研究では、BC基盤として、HYPERLEDGER FABRICを採用する。a. BCにはパブリック型とコンソーシアム型が存在し、パブリック型は不特定多数が参加できるが、コンソーシアム型は許可された特定の利用者のみが参加できる。今回は都度払いサービスを利用する物流業者や下請配送業者に限定して取引を行うため、コンソーシアム型のHYPERLEDGER FABRICを採用する。

(1)でBC上でのデータの秘匿性の課題について述べたが、その既存手法として、BCネットワークにおけるデータの共有範囲を制限する方法がある。例えば、HYPERLEDGER FABRICでは、チャンネルを定義することで、同じBCネットワークにいる他のメンバが閲覧できない、非公開の機密トランザクションを実行できるようになる[4]。しかし、物流業務では、物流業者や協力会社が配送を委託する業者は固定ではないため、受発注取引に参加する企業のみならずデータを共有するには、今後受発注取引をする可能性のある企業のみで構成される複数のチャンネルを構築しておく必要がある。また、チャンネルごとに同じ台帳を保有するため、チャンネルを利用しない場合と比較して、ノードが保有するデータ容量が増加する。従って、零細企業が保有するノードのストレージ容量が少ない場合、容量がすぐいっぱいになり、BCの書き込みができなくなってしまう可能性がある。

## 2.5 業務への適合性の要件

配送依頼から、輸配送、下請配送業者への支払の現状業務の流れについて、物流業者にヒアリングをしたところ、現状業務では配送完了のエビデンスとして出荷先の押印付き受領書が使われていることが分かった。そこで、本研究ではこの画像データを配送実績として利用する。配送完了後、ドライバ（もしくは協力会社2の事務所担当者）がモバイル端末で受領書を撮影後、対価支払迅速化システムに送信することで納品完了通知を実施し、配送実績を取得す

a HYPERLEDGER はザリナックスファウンデーションにおける登録商標です。

ることを考える。一方で、輸配送の現場においてドライバは多忙であるため、モバイル端末の操作含め、現状の物流業務への負荷が少ないようにしてほしいという意見が多かった。また、人による承認工程なしに、配送実績を対価支払迅速化システムが取得すると電子債権発行システムへの都度払い指示が自動執行される流れに抵抗感を持つ意見が多かった。そこで本研究では、ドライバの負担が少ない形で配送実績の取得ができ、かつ、配送実績や都度払いなどの承認工程を以下のように組み込むこと形で、業務に適合した業務の流れと処理フローの実現をめざす。

- (1) ドライバは配送後に、出荷先から押印付き受領書を受領後、複雑な操作なく受領書の画像を撮影し、配送実績として、画像データを対価支払迅速化システムに送信できるようにする。
- (2) 受発注業者間における、現状の押印済み受領書を送付する流れを維持できるよう、ドライバがまずは自身が所属する協力会社 2 の上長に受領書画像を送信し、上長が受領書画像を承認した後、協力会社 1、物流業者の順に受領書画像を送信できるようにする。
- (3) 発注企業は、配送実績を基に正しく納品されたことを確認し、受注企業への都度払いを承認後、都度払い指示が自動執行されるようにする。

## 2.6 関連研究

物流や貿易分野において、BCを活用して支払を自動化、迅速化する研究事例は多数報告されている。例えば Kaid らは、HYPERLEDGER FABRIC のスマートコントラクトを用いて、卸と小売業者が合意した商品が販売されたときに、自動で支払処理を実行するシステムの研究を行っている[5]。しかし、取引関係者以外へのデータの秘匿については考慮されていない。

Habib らも、BC のスマートコントラクトを活用し、商品の配送が完了すると購入者からサプライヤーと配送業者に支払処理を実行するシステムの研究に取り組んでいる[6]。しかし、上記の研究では、業務に適合した配送実績の取得や支払の実施方法に関する記載はなく、取引関係者以外へのデータの秘匿性についても考慮されていない。

また Salah らも、ETHEREUM のスマートコントラクトを活用して、購入者から販売者と配送業者への支払を自動化する仕組みについて研究している[7]。商品ごとに、配送業者と購入者用の鍵を 2 つ用意しておき、配送業者が商品を届けると、購入者と配送業者が各々の鍵を交換し、スマートコントラクトに入力することで、両者が持つ連結した文字列のハッシュ値が一致した場合、トランザクション検証に成功し支払を実行する仕組みを提案している。ただしこの仕組みでは、購入者に鍵の管理や交換などをしてもらいなど作業が発生してしまい、現状の業務フローに適合しに

くい。また、取引関係者以外へのデータの秘匿性についても考慮されていない。

Lacity らも、IoT デバイスから取得した位置情報を基に最終目的地への配達を確認し、出荷の詳細と配達の証明を検証した後に請求書を発行し、HYPERLEDGER FABRIC のスマートコントラクトに記載された支払条件に従って支払処理を実行するシステムの研究を行っている[8]。また、Kaneko らも、HYPERLEDGER FABRIC のスマートコントラクトを用いて、船荷証券の権利移転や貿易業務における振込指示を銀行のインターネットバンキングシステムと API 接続して行う研究に取り組んでいる[9]。これらの研究では、チャンネルを使用することで、秘匿性やデータの漏洩がないことを保証することを提示しているが、様々な業者間で受発注取引が発生する場合、チャンネルを複数作成する必要があり、前述した通りチャンネル作成の手間や BC のノードごとに求められるストレージ容量の限界などの課題がある。

## 3. 対価支払迅速化システムの設計

### 3.1 システム構成

図 3 に、システム構成図を示す。システム提供者と物流業者、下請配送業者は企業ごとにノードを持つ。システム提供者とは、対価支払迅速化システムを提供する企業をさす。ノードとはネットワークを構成する BC のサブシステムである。各ノードはスマートコントラクトと BC、State DB (BC と State DB をまとめて分散台帳と呼ぶ) を保持し、BC ネットワークで繋がっている。各企業は Application サーバ (以下 App サーバ) とリレーショナルデータベース (以下 RDB) を持ち、App サーバがノードの web API を呼び出し、分散台帳にデータを更新するためのトランザクションを発行する。各企業は BC ネットワークの参加者に共有するデータを分散台帳に書き込みオンチェーンで管理し、共有しないデータを各企業の RDB に置きオフチェーンで管理する。

また、対価支払迅速化システムの各企業が管理するサーバは他の企業のサーバや電子債権発行システムとのインターフェース (以下 I/F) を持つ。対価支払迅速化システムは電子債権発行システムの API を呼び出し、電子債権の発行、移転の指示を通知する。また、ドライバのモバイル端末から配送実績を取得するために、対価支払迅速化システムはモバイル端末に web アプリを配信する。

b ETHEREUM は Foundation Ethereum における登録商標です。

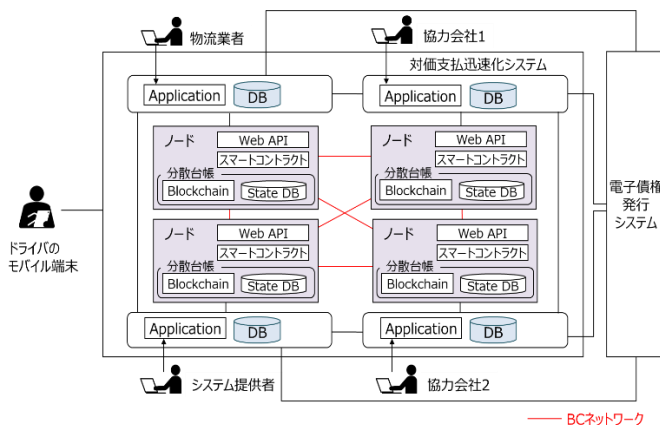


図 3 システム構成

Fig. 3 System Configuration

### 3.2 秘匿化要件の実現方法とデータの共有方法

BCを活用することの課題として、発注情報をBCで共有すると、受発注取引に関係ない企業も、発注金額などの機微情報を閲覧できてしまうことを2.4節で述べた。

そこで、本研究では、発注情報や配送実績などの生データは各企業のRDBに保存し、ハッシュ値のみを分散台帳に書き込むことで、取引に関係ない企業への秘匿性を実現する。生データの代わりにハッシュ値を利用する理由は、BCネットワークに参加する全メンバに生データのハッシュ値を共有しても、ハッシュ値から元データを復元することは困難であるため、取引に関係ない企業へのデータの秘匿性を維持できるためである。

一方で、生データをBCネットワークで保持しないため、他社に生データを共有する場合は、その他の方法で共有することが求められる。本研究では、送信者が分散台帳に書き込んだハッシュ値と受信者が送信者から受け取った生データのハッシュ値の一致を確認し、改ざんされていない生データを共有する方法を実施する。一例として、発注企業から受注企業に発注情報を送信する流れについて、図4を用いて説明する。まず、発注企業のAppサーバは、自社のRDBに発注情報の生データを格納後、ノードのweb APIを呼び出し、発注情報のKeyに紐づけてハッシュ値を分散台帳に記録する。ここでは発注情報のKeyとして、ドキュメント種別を表す文字列と種別内でユニークとなるIDを組み合わせた文字列を用いる。その後、発注企業のAppサーバは受注企業のAppサーバに発注情報の生データとKeyを送信し、受注企業のAppサーバは受け取ったKeyに紐づけて、分散台帳からValueを検索し、ハッシュ値を読み出す。HYPERLEDGER FABRICのState DBのデータ構造はKey-valueであり、これを活用することで、ハッシュ値の検索が可能となる。その後、受注企業のAppサーバは発注企業から受け取った発注情報から算出したハッシュ値と、分散台帳から読み出したハッシュ値を照合し、同じ値であることを確認後、受注企業のRDBに発注情報の生データを格納する。最後に、発注情報を受け取ったこと的成功レスポ

スを発注企業のAppサーバに送信する。

- ① 発注情報の生データのRDB保存
- ② 発注情報のハッシュ書込
- ③ 発注情報のハッシュ書込完了確認
- ④ 発注情報の生データ、keyの送付
- ⑤ 発注情報のハッシュ値読み出し
- ⑥ ハッシュ値の値を照合
- ⑦ 発注情報の生データのRDB保存
- ⑧ 発注情報の受信返答

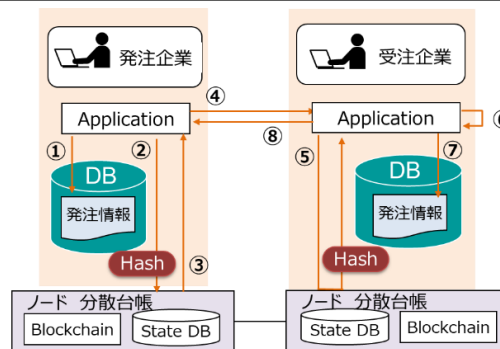


図 4 発注情報送信の流れ

Fig. 4 Flow of sending order information

### 3.3 配送実績の取得方法と承認工程の組み込み

2.5節では、業務への適合性の要件について述べた。本節では、対価支払迅速化システム利用時の業務フローが現状の業務に適合できるような配送実績の取得方法と配送実績と都度払いの承認工程の実現方法について説明する。

#### 3.3.1 配送実績の取得と承認の流れ

図5を用いて、ドライバがモバイル端末を用いて受領書画像を撮影し、対価支払迅速化システムが配送実績を取得する流れを説明する。前述した通り、輸配送の現場においてドライバは多忙であり、ドライバに複雑な端末操作を強いることは難しい。そのため、ドライバが煩雑な操作をせずに、受領書画像を撮影し、送信できる方法が必要である。複雑な操作の例としては、ドライバが端末のカメラアプリで撮影した画像をメールに添付して直属の上長に送信するなどがあり、画像を受け取った上長も画像の管理が煩雑になるため、このような方法は避ける必要がある。そこで本研究では、以下の手順に沿って受領書の配布、配送実績の取得を行うことで、ドライバへの負荷が少ない形で受領書画像を取得する方法を実現する。

1. 二次元コード付き受領書の受け取り：物流業者はドライバによる集荷時に、対価支払迅速化システムから二次元コード付き受領書を印刷し、ドライバに配布する。
2. web アプリの起動：配送完了後、1.で物流業者から受け取った受領書に出荷先が押印後、ドライバはモバイル端末を用いて、受領書の二次元コードを読み取る。二次元コードにはwebアプリを配信する物流業者のAppサーバのURLが埋め込まれており、モバイル端末のブラウザ上でURLを開き、ブラウザからサーバにリクエストを送ることで、webアプリが起動する。ここでは、現状の業務において、ドライバに配布するモバイル端末にネイティブアプリをダウンロードできない場合を想定し、webアプリで受領書画像の撮影、送信ができるようにする。

3. 受領書画像の撮影: ドライバのモバイル端末上で web アプリを起動した後、画面上に表示されるガイドに従って、ドライバは受領書画像の撮影と送信を行う。ここで、適切な撮影範囲が収まった受領書画像を撮影できるよう、web アプリに次の機能を構築する。その機能は、二次元コードから取得した画面上における二次元コードの座標情報とサーバから取得した紙面上における撮影範囲の座標情報から、画面上の撮影範囲の座標情報を推定し、適切な撮影範囲に受領書が収まると自動でシャッターが切れるという機能である。その後、撮影された受領書画像が問題なければ、ドライバは送信ボタンを押す。

4. 受領書画像の送信と他の企業への共有: ドライバが送信ボタンを押下後、受領書画像の二次元コードに埋め込まれた配送 ID と紐づけて、ドライバのモバイル端末から末端の下請配送業者である協力会社 2 の App サーバに画像データが送信される。その後、ドライバの上長が PC などのクライアント端末で画像を確認後、内容に問題がなければ承認ボタンを押す。その後、協力会社 2 の App サーバが受領書画像の生データを RDB に保存し、ハッシュ値のみを BC に書き込む。ここで初めて受領書画像が配送実績として対価支払迅速化システムに登録される。その後、3.2 節に記載した生データの共有方法と同様の方法で、耐改ざん性を維持しつつ、協力会社 2 から協力会社 1、協力会社 1 から物流業者へと順番に受領書画像が送信される。

これにより、ドライバは受領書の二次元コードを端末で読み取り、web アプリのガイドに従うだけで、容易に受領書画像の撮影と送信を実施できる。また、現状の受領書の承認工程を維持したまま、配送実績を物流業者と下請配送業者間で共有することが可能となる。

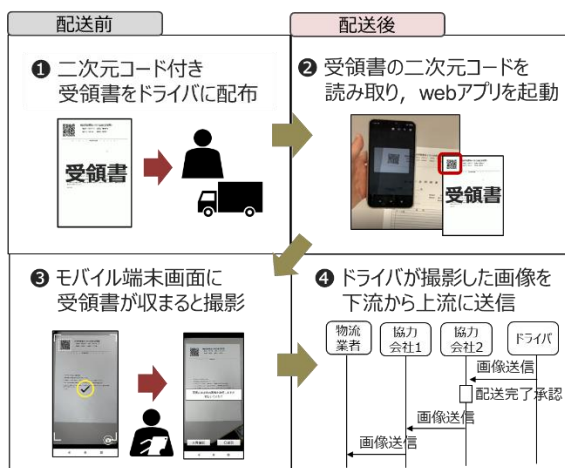


図 5 受領書画像の送信方法

Fig. 5 How to send delivery note image

### 3.3.2 都度払いの承認と月末決済の流れ

図 6 に、都度払い承認の流れと月末に物流業者が金融機関に行く債務引受代金の決済の流れを説明する。発注情報に紐づく全配送実績が登録されると、対価支払迅速化システムは協力会社 1 から物流業者への請求書を自動発行する。

物流業者が配送実績と請求書を確認し、協力会社 1 への都度払いの承認ボタンを押すと、対価支払迅速化システムが電子債権発行システムの API を呼び出し、電子債権の発行と協力会社 1 への移転が実行される。その後、対価支払迅速化システムは電子債権発行システムから成功レスポンスを受け取ると、協力会社 1 から物流業者への領収書と協力会社 2 から協力会社 1 への請求書を自動発行する。協力会社 1 はその領収書と請求書、配送実績を確認後、協力会社 2 への都度払いの承認ボタンを押す。その後、同様の手順で協力会社 2 への電子債権の移転が実行される。その後、任意のタイミングで下請配送業者が、電子債権発行システム上で電子債権を現金に変換するための現金化申請を行うと、金融機関から下請配送業者の指定の口座に現金が振り込まれる。

つぎに、物流業者から金融機関への月末決済の流れを説明する。金融機関は物流業者の債務を代行しているため、定期的なタイミングで物流業者から金融機関に債務引受代金の決済をする必要がある。物流業者のキャッシュアウトが月末の場合、月末に物流業者は債務引受代金を対価支払迅速化システム上で確認し、金融機関への決済を実行する。また、この時点で現金化されていない電子債権は自動的に下請配送業者の口座に振り込まれる。

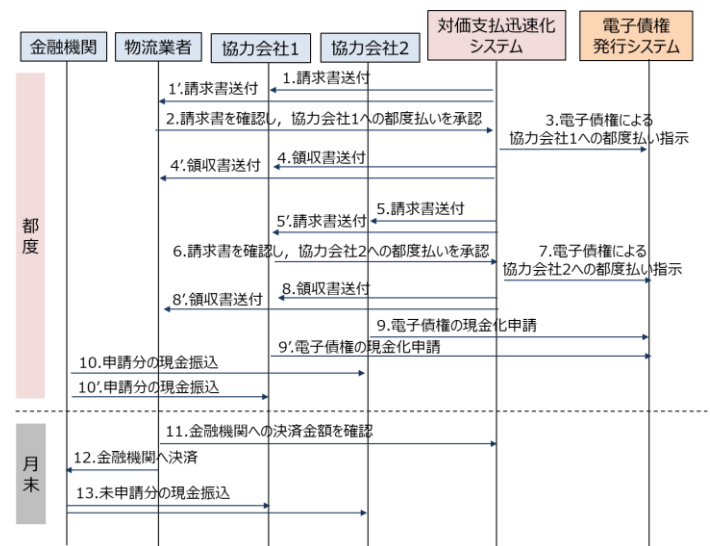


図 6 都度払いと月末決済の流れ

Fig. 6 Flow of payment each time and month-end settlement

## 4. 実験内容と評価

### 4.1 実証実験用システムのネットワーク構成

図 7 に示すネットワーク構成において、実証実験用の対価支払迅速化システムの開発を行った。各企業の App サーバ、RDB、ノードはそれぞれ異なる AMAZON WEB SERVICES (以下 AWS と略す) のインスタンス上で動作す

るc. ノード間の通信は AWS の VPC 内のプライベートネットワーク上で行う。本研究では、物流業者が1社、協力会社1が5社、協力会社2が10社いる想定でインスタンスを構成した。BC 基盤としては HYPERLEDGER FABRIC V2.2 を用いた。また、AWS のインスタンス上のサーバ OS として UBUNTU 16.04.6 LTS を構築したd。

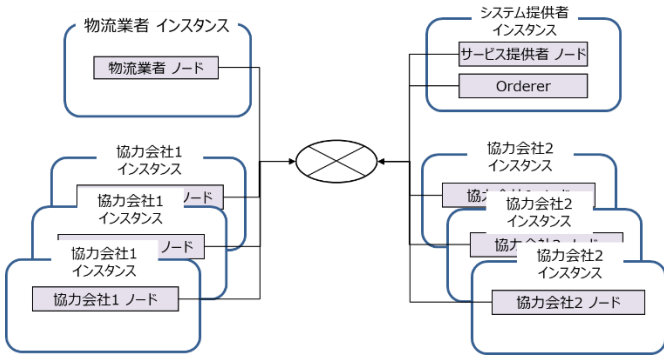


図 7 ネットワーク構成

Fig. 7 Network Configuration

#### 4.2 秘匿性と書き込みデータ量削減の妥当性検証

本研究では、機微情報の秘匿要件を満たすために、発注情報や配送実績などの生データは各企業の RDB に保存し、ハッシュ値のみを分散台帳に書き込むことで、取引に関係ない企業への秘匿性を維持できることを提示した。本節では、受発注業者間で共有する文書と画像の生データを分散台帳に書き込む場合と、ハッシュ値を分散台帳に書き込む場合のデータサイズを比較し、分散台帳への書き込みデータ量の削減を検証する。

実証実験用システムでは、つぎの文書データと画像データを企業間で共有する。1) 発注情報、2) 都度払いの請求書、3) 都度払いの領収書、4) 受領書画像。1), 2), 3)は物流業者と協力会社1で取引される文書と、協力会社1と協力会社2間で取引される文書が発注ごとに存在する。また4)は配送ごとに画像が存在する。本研究では、各文書データについては、ハッシュ関数 SHA-256 を用いて、JSON オブジェクトの文字列データからハッシュ値を算出した。画像データについては、ハッシュ関数 MD5 を用いて算出したハッシュ値を活用した。

表 1 に、分散台帳に生データを書き込む場合とハッシュ値を書き込む場合のデータサイズを比較した結果を示す。これは、1回の発注で5件の配送依頼をする場合における、各文書の生データとハッシュ値のデータサイズである。ハッシュ値のデータサイズは生データのサイズに関係なく、ハッシュ関数 SHA-256 を用いて算出する場合、32Byte となり、ハッシュ関数 MD5 を用いて算出する場合、16Byte となる。発注情報の生データは、1配送が追加されると450Byte ずつデータサイズが増加する。その他の文書データのサイ

ズは配送数に関係なく一定である。

1回の発注で5件の配送依頼をする場合、1発注で削減できるデータ量を算出する。生データを分散台帳に書き込む場合は、1)発注情報と2)請求書、3)領収書は1回、4)受領書画像は5回書き込まれる。一方、ハッシュ値を分散台帳に書き込む場合は、物流業者と協力会社1間で送受信する際に1回書き込み、協力会社1と協力会社2間で送信する時に1回書き込まれるため、1発注につき、1)発注情報と2)請求書、3)領収書は2回、4)受領書画像は5回配送あるため10回書き込まれる。従って、生データを分散台帳に書き込む場合、 $2.3+0.07+0.26+1600*5=8002.63(\text{KB})$ のデータサイズとなる。一方ハッシュ値を書き込む場合のデータサイズは、 $0.032*2+0.032*2+0.032*2+0.016*10=0.352(\text{KB})$ となり、1発注事に8002.278(KB)のデータサイズを削減できる見込みを得た。

検証結果より、ハッシュ値を書き込むことで、分散台帳に書き込むデータサイズを削減できることが分かった。従って、生データの代わりにハッシュ値を分散台帳に書き込むことで機微情報の秘匿化を実現できるとともに、生データの代わりにハッシュ値を書き込むことでデータサイズを削減できるため、複数の文書ファイルや画像データのハッシュ値をまとめて一つのブロックに書き込むことで、ブロックの数を削減でき、各企業のノードのストレージの使用量を削減できる見込みを得た。

表 1 分散台帳の書き込みデータ量の結果

Table 1 Result of writing data size on distributed ledger

データ種別	分散台帳の書き込みデータ量 (KB/1ファイル)	
	生データの場合	ハッシュ値の場合
発注情報	2.3	0.032
請求書	0.07	0.032
領収書	0.26	0.032
画像	1600	0.016

#### 4.3 現場担当者へのヒアリング結果

物流業者と下請配送業者の経理担当者と配車担当者数名に、実証実験用システムを用いて、輸配送代金の都度払いのデモを行い、業務への適合性についてヒアリングを実施した。具体的には、発注情報の取得から受領書画像の撮影、都度払い、月末決済までの流れについて説明した。デモに加えて、画面操作も体験してもらった。

現場の担当者からは、システムの使い勝手はよく、業務への適合性に関しても大きな問題はないとご意見を頂いた。また、都度払いの度に発行される請求書や領収書を管理し、毎月の銀行への決済金額を算出するのが大変な印象があっ

c AMAZON WEB SERVICES はアマゾンテクノロジーズインコーポレイテッドにおける登録商標です。

d UBUNTU はキャノニカルリミテッドにおける登録商標です。

たが、都度払いした金額が自動集計されて表示されるのであれば特に問題はないとご意見頂けた。一方で、下請配送業者が都度払いサービスを利用するかどうかで、物流業者側の月末支払の業務が煩雑にならないよう、支払手順を一本化してほしいなどのご意見も頂いた。

また、ドライバがモバイル端末で撮影した受領書画像を配送実績として取得する仕組みに関しても意見をもらった。都度払いを承認する前に受領書画像を確認できる仕組みや正しい撮影範囲が収まった受領書画像を撮影できるようにする仕組みについては賛同頂けた。一方で、出荷先が発行した受領書を利用する場合における、受領書の二次元コードの印字方法や二次元コードの受領書が複数枚存在する場合にどのように撮影するかなどの意見もあった。また、物流業者からは各企業がBCのノードを持っていない場合に、BCの構成をどのようにするか等の指摘もあった。

物流業者や下請配送業者へのデモとヒアリングを通じ、システムの使い勝手はよく、都度払いサービス利用時の全体の業務フローについても、物流現場のオペレーション上大きな問題はないことを確認できた。一方で、頂いた指摘事項に関しては、今後対策を検討する。

## 5. まとめ

配送業者の資金繰り改善のため、BCを活用した対価支払迅速化システムの開発を行った。BCを活用することで、発注情報や配送実績の透明性や耐改ざん性を保証できる一方、受発注取引に関係のない企業への機微情報の秘匿性を維持する必要があった。また、現状の物流業務に沿った処理フローの流れが求められた。具体的には、対価支払迅速化システムに配送実績を登録する前や都度払いを実施する前に人が承認するフローを入れることや、ドライバの現状の物流業務への負荷が少ない形で配送実績を取得することが求められた。本研究では、BCに生データを書き込む代わりにハッシュ値を書き込むことで、秘匿性を維持しつつ、分散台帳に書き込むデータサイズを削減できることを示した。また、物流業者や配送業者に都度払いのデモを実施し、現状業務への適合性を示すことができた。これにより、対価支払迅速化システムを用いることで、機微情報の秘匿性を維持しつつ、現状業務の流れに沿った形で下請配送業者への都度払いを実施でき、物流業者から配送業者への支払サイトを、数週間から1日に短縮できる見込みを得られた。

今後は、下請配送業者が都度払いサービスを利用するかどうかで、物流業者側の月末支払の業務が煩雑にならないよう、支払手順を一本化する仕組みを検討するとともに、受領書が複数枚存在する場合などにおける配送実績の取得方法を模索する。また、今回のBCの構成は各企業がノードを一つ持つ構成にしたが、各企業がノードを持っていない場合に、機微情報の秘匿要件を維持する方法についても検討する。

**謝辞** ヒアリングにご協力頂いた物流業者、配送業者の皆様、本研究をご支援頂いた皆様に、謹んで感謝の意を表する。

## 参考文献

- [1] 国土交通省：「トラック運送業における下請・荷主適正取引推進ガイドライン」, 入手先 ([https://www.mlit.go.jp/kisha/kisha08/09/090314\\_2/01.pdf](https://www.mlit.go.jp/kisha/kisha08/09/090314_2/01.pdf)) (21/12/7 閲覧)
- [2] DHL：“Blockchain in Logistics” (2018)
- [3] 国土交通省：「令和2年度 宅配便取扱実績について」, 入手先 (<https://www.mlit.go.jp/report/press/content/001418259.pdf>) (21/12/7 閲覧)
- [4] E. Androukaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A.D. Caro, ..., J. Yellick：“HYPERLEDGER FABRIC: a distributed operating system for permissioned blockchains”, Proc. of EuroSys '18, Vol.30, pp.1-15 (2018)
- [5] D.kaid, M.M.Eljazzar：“Applying Blockchain to Automate Installments Payment between Supply Chain Parties”, Proc. of 2018 14th International Computer Engineering Conference, pp.231-235, (2018)
- [6] M.A.Habib, M.B.Sardar, S.Jabbar, C.M.N.Faisal, N. Mahmood, M.Ahmad：“Blockchain-based Supply Chain for the Automation of Transaction Process: Case Study based Validation”, Proc. of 2020 International Conference on Engineering and Emerging Technologies, pp.1-7 (2020)
- [7] K.Salah, H.Hasan：“Blockchain-based Solution for Proof of Delivery of Physical Assets”, Blockchain – ICBC 2018, pp 139-152, (2018)
- [8] M.C.Lacity, R.V.Hoek：“How Walmart Canada Used Blockchain Technology to Reimagine Freight Invoice Processing”, MIS Quarterly Executive, Vol. 20 : Iss. 3 , Article 5
- [9] 金子 雄介, 田村 浩気, 河合 伸浩, 田中 俊太郎, 岡 知博：“貿易実務のブロックチェーン利用, 実践と課題”, デジタルプラクティス, Vol.10 No.3, pp 492 - 505