

# ブロックチェーン時代の 簿記システム

板橋雄大

# 1.概要

ブロックチェーン（BC）技術（および、分散型台帳技術；DLT）の一般的特徴

①連結されるブロック同士の時系列連続性

ブロックチェーン（BC）技術（および、分散型台帳技術；DLT）は、取引の記録されたブロックが時系列的に連続した形で順次追加されていくという特徴を持っている。

②ブロックに格納されているデータに対する一定の信頼性→ブロックを新しく追加するたがめには、新規のブロックがネットワークの参加者間での合意形成を経る必要がある。連結されたブロックに格納されているデータにはネットワークの参加者間での合意形成を経たものであるという一定の信頼性が担保される。

③改竄耐性→追加されるブロックには、取引に係るデータだけでなく、ひとつ前のブロックを暗号化したハッシュ値も保存されており、これによってブロック間の時系列的順序が設定される。この結果、過去の特定のブロックに含まれる情報を改竄するたためには、そのブロックの情報を引き継ぐ以降のブロックすべてを、ネットワークの他のノードよりも早く全て改竄する必要があるため、改竄耐性に優れているという特徴もある。

④分散性と(不完全ではあるが)即時性&⑤取引履歴の追跡可能性・検証可能性

→ネットワークの参加者がそれぞれデータのコピーを持ち、分散して保存、管理する仕組みであるため、ブロックチェーン上の情報は、全ての参加者によって閲覧可能で、変更履歴も追跡できるという、透明性・検証可能性という特徴も持っている。

# 1. ブロックチェーンを簿記システムとして利用 する際の問題点

このような特徴を持つBC技術を簿記システムに応用することが出来れば、より高いセキュリティ性、透明性、効率性、信頼性を持った形で、「営業開始のときから決算期にいたる一さいの財産の変動について（中略）複式簿記の方法で『取引の歴史記録・取引の勘定分解記録・取引の勘定分類記録・取引の明細記録』を設け」（片野，1975, 166）ることが出来ると考えられる。

しかしながら、現行のBC技術の多くは、暗号資産の取引を主として利用されていることもあり、パブリック型のBCネットワークの機能は一般的な簿記システムに用いるには不十分な部分がある。

例えば、

「①連結されるブロック同士の時系列連続性」「⑤取引履歴の追跡可能性・検証可能性」については、取引資産の分割や統合が起こった場合、それより以前の情報の追跡は困難である。

「②ブロックに格納されているデータに対する一定の信頼性」、「③改竄耐性」という特徴は、既に知られている通り、ネットワークに参加しているメンバーを信頼できない環境では（合意形成プロセスの種類によっては）破られることがある。

「④分散性と（不完全ではあるが）即時性」は、どこまでの範囲の情報利用者に提供するかという問題へとつながる。企業の取り扱う簿記データには取引価格などの情報が含まれる以上、これらを無制限に提供する事はできない。

一方で、プライベート型については、④については取引を複数のチャンネルで行うことで、自分が所属しているチャンネル以外に取引について見ることが出来ないようにすることができる。また、同一チャンネル内においても、取引を秘匿することも出来る。

②③については、プライベート型のBCは管理者が存在することによって、ネットワークへの参加者を限定し、管理することが出来る。

①②については、パブリック型と同じ問題が発生するが、分割や結合の際に元の資産を示す「属性情報」を付与することによって、問題を解決することが出来ると考える。

## 2.DLTの種類

- 管理者に許可された単一或いは複数の限定的なノードが取引の検証・承認を担う行つたものをPermissioned型、許可が不要なものをPermissionless型に分類される。
- 台帳が誰でもアクセス可能か、ネットワークの許可された参加ノードのみがアクセス可能かによって、パブリック型とプライベート型に分類される。
  - パブリック型が誰でも許可なくネットワークに参加でき、管理者が存在しないのに対して、プライベート型は特定の管理者が存在し、管理者によって許可された者のみがネットワークに参加できるという違いがある。
- パブリック型の利点
  - ネットワーク内で情報が完全に共有されることで、いわゆる透明性が高いことと、中央集権的な管理者が存在しないためネットワークに発生する障害に対して耐性がある(システムを構成するハードウェア群において、そこが故障することでシステム全体が停止するような単一障害点が存在しない)ことが挙げられる。
- プライベート型の利点
  - 取引データが外部に公開されないためプライバシーが確保されることと、ノードの数が少なく、かつ各ノードの参加を管理者が承認しており信頼性の低いノードの存在を想定しなくて済むことから、取引の承認のための合意形成にかかる時間が短く大量の取引データの処理が迅速に行える点が挙げられる。

2-1: 米国トップ100社の採用している主要な3つのDLTプラットフォーム (Blockdata,2021,p.15) とそ

	Hyperledger Fabric	Ethereum	Quorum
ネットワークへの参加許可	Permissioned	Permissionless	Permissioned
管理者の有無	管理者有	管理者無	管理者無
取引の追跡可能性	取引データを共有したいノードを含むチャンネルを作成する形式。そのためチャンネル内のノードは全ての情報を参照可能。	ネットワーク内の全ノードはブロックデータを共有しているため、追跡が可能。	暗号化によって取引の内容の秘匿化のみ可能。PMTs(privacy marker transactions)を設定する場合、取引の存在自体を非公開にできる。
取引主体の開示	一般的にはチャンネル内のノードは全トランザクションを共有し、取引主体を確認できるが、private data設定により一部のノードのみが確認できるようにも設定可能。	トランザクションに含まれる情報に関しては匿名性無し。	取引に係る主体の電子的アドレスは共有される。PMTsを設定し取引データをBCから外す場合、取引の存在自体を非公開にできる。
合意形成	RAFT /KAFKA(非推奨)	POW(PoSも試験導入)	Clique POA(非推奨)/ RAFT(非推奨)/BFT
処理スピード	速い	遅い	中間

### 3. 実践システム概要

本研究では、企業向けのBCプラットフォームとして利用例の多いHyperledger Fabric(HF)を用いて、ブロックチェーン時代の簿記システムについて検討したい。

HFは、ブロックチェーンが実現する分散台帳を金融やサプライチェーンだけでなく幅広い企業や団体のビジネスシーンで活動できるように改善することを目的として2015年12月に開始されたHyperledgerプロジェクトの1つのプロジェクト名である。

Hyperledgerプロジェクトは、オープンソースであり、全てのソースコードが公開されている。そのため、提供されているプログラムを簿記システムに適した形に修正することが可能である。

# 3 - 1 : サンプルプログラムの修正内容

今回使用したメインプログラムは、「Fabric test network(以下、test network)」というサンプルプログラムで、これはローカルPC上でFabricについて学習・稼働検証を行う目的でhyperledger-fabric webサイトにおいて提供されているものである。

test networkにはAssetの創出、転送、情報の更新、消滅を行う機能が備わっているが、これを本研究の目的に沿って、Assetに加えliability、Equity、Profit、Lossというカテゴリーを追加し、Asset同様に創出、転送、情報の更新、消滅の処理ができるようにスマートコントラクトの機能の追加を行っている。

(なお、ブロックに記録された過去データを消滅させることはブロックチェーンの性質上できないので、本来サンプルプログラムでは、消滅処理においては、アイテムのID以外の属性を全て空白にしたものを新しく記録する形で行われている。)

また、詳細は後述するが、各カテゴリーのアイテムには新しく、「消滅フラグ」、「取引日」「（チャンネル全体で一意に定められた）追跡管理用の番号」、「資産・負債・純資産・費用・収益のどのカテゴリーに所属するか」、「登録名称」、「所有者」、「金額」、「創出時取引相手項目追跡番号」、「消滅時取引相手項目追跡番号」といった属性情報を付与した。

- 消滅フラグはそのアイテムが消滅しているか否かを記録するためのタグである。本来のメインプログラムでは消滅機能は、アイテムの全ての属性情報を消し、BCネットワークの最新状態についての検索を行っても、そのアイテムが表示されなくするものであった。
- そこで本研究では、消滅処理において、各属性情報はそのまま残し、消滅状態であることを消滅フラグを通じて示すように変更した。この機能変更と、「創出時取引相手項目追跡番号」、「消滅時取引相手項目追跡番号」の2つの属性を付与したことで、取引データの追跡・検証機能を改善した。

- 属性情報は理論上無限に設定することができるため、「経済活動には活動時点、活動業種(営業、投資、財務など)、活動場所(部門、地域など)、活動責任者などといった属性」(石川純治,1994,p.120)、そして、「商品在庫であればそれを識別する商品番号、在庫量、取得原価、時価、保管場所、管理責任者など多くの属性」(同上)を設定することが出来る。
- ただし、入力すべき属性が増えれば増える程、入力の際の手間は増え、人為的ミスが生じる危険性も増加する。そのため、例えば経済活動に関していえば、その入力を行う機器情報を自動的に付与するような仕組みを用いて、出来る限り自動化する必要があるだろう。
- また、時価情報については、重要な属性であるが、これをブロックチェーン上で日々記録してゆくのは、ネットワークのパフォーマンスを低下させ、場合によってはトランザクションの中断を引き起こすこととなる。

(特に、時価情報をブロックチェーン上で処理しようとする時、それぞれのネットワーク参加者は、その時価情報が正しいのかを自分自身でも処理する必要がある。外部データベースへのアクセス結果が、アクセスするタイミングによって異なる場合は合意形成に失敗することとなるし、計算式による現在価値の計算も同様の問題を引き起こす。)

- したがって、本研究では入力によって時価情報を更新してゆくのではなく、時価情報は出力において処理することが望ましいと考えられる。

# 1. ブロックチェーンを簿記システムとして利用 する利点

- 属性情報は、プラットフォームのインストール時（ブロックチェーンの立ち上げ時）にAssetにはどのような属性があるのかを事前に設定しておく必要がある。
- その上で、BCでの取引の指示を入力する時には求められる属性を入力する必要がある。
- 例えば、下記のように行う。

```
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile
${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tl
sca.example.com-cert.pem -C bookkeepingchannel -n basic --peerAddresses localhost:7051 --
tlsRootCertFiles
${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --
peerAddresses localhost:9051 --tlsRootCertFiles
${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt -c
{"function": "CreateOtherAsset", "Args": ["20240801", "B00001", "XyzDeposit1", "Xyz", "500000", "1",
"B00002", ""]}
```

赤字以外の部分は、ネットワークに参加している各メンバーがトランザクションの承認を自動的に行わせるための入力情報である。以降の説明においてはこの部分は省略する。

入力された情報は、プラットフォームのインストール時に設定した情報と照合され、「2024年8月1日」が取引日、「B0001」が追跡管理用の番号「XyzDeposit1」が登録名称「Xyz」が所有者、「500000」が取得単価、「1」が個数、B00002はこのアイテムの発生原因となる取引相手項目追跡番号である。消滅時取引相手項目追跡番号は現時点では不明のため、「データ無し」というアイテムとして創出され管理される。

なお、ここで消滅フラグ及びカテゴリ属性については入力していないが、これはCreateをするということは消滅していないことは明らかであり、またOtherAssetを作成する指示である以上カテゴリーはOtherAssetになるためである。このように、自明の属性については、出来る限りシステムが自動的に入力するようにすることで、入力時の人為的ミスを減少させることが可能となる。

実際に記録されているかを確認するために、ReadOtherAssetコマンドを実行した。これはOtherAssetとしてカテゴライズされたアイテムのIDを指定することで、その記録データを表示するものである。下図のように、先ほど入力した内容がブロックチェーン上に記録されていることが分かる。

```
/peer0.org2.example.com/tls/ca.crt -c '{"function":"ReadOtherAsset","Args":["B00001"]}'  
,  
2024-08-30 17:11:11.896 JST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200 payload:"{\n\"DeletedFlag\":0,\n\"Date\":\n\"20240801\", \n\"ID\":\n\"B00001\", \n\"Type\":\n\"OtherAsset\", \n\"Name\":\n\"XyzDeposit1\", \n\"Owner\":\n\"Xyz\", \n\"Price\":500000, \n\"Quantity\":1, \n\"Inid\":\n\"B00002\", \n\"Outid\":\n\"\"}"
```

同様にして、資本アイテムの創出、

```
{"function":"CreateCapital","Args":["20240801", "B00002", "XyzCapital1", "Xyz", "500000", "1", "B00001", ""]}
```

資産アイテムの創出

```
{"function":"CreateOtherAsset","Args":["20240803", "B00003", "XyzDeposit2", "Xyz", "300000", "1", "B00004", ""]}
```

負債アイテムの創出

```
{"function":"CreateLiability","Args":["20240803", "B00004", "XyzLiability1", "Xyz", "300000", "1", "B00003", ""]}
```

を行った。ブロックチェーンでは、複式簿記の構造をそのまま再現することは困難であるため、このように、分離して処理を行っている。

既に存在するB00001アイテムについて、このあとの支払いに用いるために、当該支払いに適切な金額を分離することとなる。その前段階の処理として「消滅時取引相手項目追跡番号」をUpdateによって記録する。

```
{"function":"UpdateOtherAsset","Args":["20240801", "B00001", "XyzDeposit1", "Xyz", "500000", "1", "B00002", "sundries"]}
```

「消滅時取引相手項目追跡番号」を記録した上で、  
{ "function": "DeleteOtherAsset", "Args": ["B00001"] }

として、アイテムを消滅させる（属性情報はそのまま、消滅フラグを0から1へと変更している）。

支払用に適切な金額の資産を改めて創出する。

```
{ "function": "CreateOtherAsset", "Args": ["20240809", "B00005",  
"XyzDeposit3", "Xyz", "100000", "1", "B00001", "B00007"] }
```

```
{ "function": "CreateOtherAsset", "Args": ["20240809", "B00006",  
"XyzDeposit4", "Xyz", "400000", "1", "B00001", ""] }
```

この2つのコマンドを実行したことで500,000円の資産を100,000円と、400,000円に分離することが出来たことになる。

このうちB00005については、Pt社への支払に用いるので、転送コマンドで所有者をPtに変更する。

```
{ "function": "TransferOtherAsset", "Args": ["B00005", "Pt"] }
```

支払が行われたため、本来はPt社側で資産をブロックチェーン上に創出した上で、Xyz社（当社）宛に転送する処理が必要だが、冗長となるため、割愛し、Xyz者側で直接、創出処理を行う。

```
{"function":"CreateOtherAsset","Args":["20240809","B00007","XyzInventory1","Xyz","100000","1","B00005",""]}
```

次の支払の準備のため、**消滅時取引相手項目追跡番号**を付与し、

```
{"function":"UpdateOtherAsset","Args":["20240809","B00006","XyzDeposit4","Xyz","400000","1","B00001","sundries"]}
```

その上で、資金を支払額に分離するために、資産を消滅させ、

```
{"function":"DeleteOtherAsset","Args":["B00006"]}
```

分離した金額で2つの資産を創出する。

```
{"function":"CreateOtherAsset","Args":["20240810","B00008","XyzDeposit5","Xyz","200000","1","B00006","B00010"]}
```

```
{"function":"CreateOtherAsset","Args":["20240810","B00009","XyzDeposit6","Xyz","200000","1","B00006",""]}
```

(以下割愛)

最後に、減価償却資産の創出を行う。このアイテムはこれまでのアイテムと異なり、償却率(0.1)が属性として付与されている。

```
{"function":"CreatePresentvalueCalcableAsset","Args":["20240810",  
"B00010", "Xyzdepreciableassets1", "Xyz", "200000", "1", "B00008", "",  
"0.1"]}
```

また売買目的金融商品として、B00013を創出した。このアイテムには、3675が取得単価、50は数量、13010が証券コードである。なお、証券コード1301は極洋株式会社であるが、外部データベースへのアクセスの都合上、このようになっている。

(ブロックチェーンからではなく、後述するwebアプリケーションからアクセスする)

```
{"function":"CreateFinancialAsset","Args":["20240810", "B00013",  
"Xyzfinancialassets1", "Xyz", "3675", "50", "B00011", "", "13010"]}
```

株

- ここで、BCネットワークから出力したデータが下記のような状態であった。
- {"DeletedFlag":0,"Date":"20240810","ID":"B00013","Type":"FinancialAsset","Name":"Xyzfinancialassets1","Owner":"Xyz","Price":3675,"Quantity":50,"Inid":"B00011","Outid":"","StockNumber":"13010","PresentValue":183750}
- {"DeletedFlag":0,"Date":"20240810","ID":"B00010","Type":"PresentvalueCalcableAsset","Name":"Xyzdepreciableassets1","Owner":"Xyz","Price":200000,"Quantity":1,"Inid":"B00008","Outid":"","Depreciationrate":0.1,"PresentValue":200000}
- {"DeletedFlag":1,"Date":"20240801","ID":"B00001","Type":"OtherAsset","Name":"XyzDeposit1","Owner":"Xyz","Price":500000,"Quantity":1,"Inid":"B00002","Outid":"sundries"}
- {"DeletedFlag":0,"Date":"20240803","ID":"B00003","Type":"OtherAsset","Name":"XyzDeposit2","Owner":"Xyz","Price":300000,"Quantity":1,"Inid":"B00004","Outid":""}
- {"DeletedFlag":0,"Date":"20240809","ID":"B00005","Type":"OtherAsset","Name":"XyzDeposit3","Owner":"Pt","Price":100000,"Quantity":1,"Inid":"B00001","Outid":"B00007"}
- {"DeletedFlag":1,"Date":"20240809","ID":"B00006","Type":"OtherAsset","Name":"XyzDeposit4","Owner":"Xyz","Price":400000,"Quantity":1,"Inid":"B00001","Outid":"sundries"}
- {"DeletedFlag":0,"Date":"20240809","ID":"B00007","Type":"OtherAsset","Name":"XyzInventory1","Owner":"Xyz","Price":100000,"Quantity":1,"Inid":"B00005","Outid":""}
- {"DeletedFlag":0,"Date":"20240810","ID":"B00008","Type":"OtherAsset","Name":"XyzDeposit5","Owner":"Dt","Price":200000,"Quantity":1,"Inid":"B00006","Outid":"B00010"}
- {"DeletedFlag":1,"Date":"20240810","ID":"B00009","Type":"OtherAsset","Name":"XyzDeposit6","Owner":"Xyz","Price":200000,"Quantity":1,"Inid":"B00006","Outid":"sundries"}
- {"DeletedFlag":1,"Date":"20240810","ID":"B00011","Type":"OtherAsset","Name":"XyzDeposit5","Owner":"Xyz","Price":183750,"Quantity":1,"Inid":"B00009","Outid":"B00013"}
- {"DeletedFlag":0,"Date":"20240810","ID":"B00012","Type":"OtherAsset","Name":"XyzDeposit6","Owner":"Xyz","Price":16250,"Quantity":1,"Inid":"B00009","Outid":""}
- {"DeletedFlag":0,"Date":"20240803","ID":"B00004","Type":"Liability","Name":"XyzLiability1","Owner":"Xyz","Price":300000,"Quantity":1,"Inid":"B00003","Outid":""}
- {"DeletedFlag":0,"Date":"20240801","ID":"B00002","Type":"Capital","Name":"XyzCapital1","Owner":"Xyz","Price":500000,"Quantity":1,"Inid":"B00001","Outid":""}
- この中で、**Xyzfinancialassets1**、**Xyzdepreciableassets1**についてはそれぞれ**183,750円(3675円×50株)**、**200,000円**で取得していることが分かる。また、時価の算出については、そのために必要な属性は記録されているもののブロックチェーン内では算出は行っていないこともわかる。
- Javascriptによって作成したwebブラウザ上でこれらのデータを読み込むことで、表示時点の時価を表示できるようにした結果が次のスライドである。

# DeleteFlagによる追跡可能性の変化

本来はDeleteによって消滅処理がされたアイテムは、ブロックチェーンの現在の状態を表示する際には表示されなくなる。（過去記録の表示は可能）

つまり、記録に抜けがある状態で表示される。

仮に、一定の期間ごとに現在状態を出力し、保存したとしても、短期間に創出から消滅までを繰り返すようなことがあると、そこで追跡可能性は途切れてしまう。

一方、Flagによる処理（存在状態で0、消滅状態で1の値を取る）では、すべての記録情報を取得することができる。

追跡可能性に関するもう一つの問題は、例えば「創出時取引相手項目追跡番号」「消滅時取引相手項目追跡番号」のどちらかだけを利用する場合に発生する。前述のよつに、属性の数は、ブロックチェーンの立ち上げ時に設定する必要がある。したがって、「創出時取引相手項目追跡番号」を1相手項目のみに設定していると、2つ以上のアイテムが取引相手項目となった場合に、追跡が出来なくなる。

複数のアイテムを集めて新しいアイテムが創出される場合も、複数のアイテムへと分離され新しいアイテムが創出される場合も、両方ともに対応するためには、「創出時取引相手項目追跡番号」、「消滅時取引相手項目追跡番号」の2つが必要であると考えられる。

- 消滅させる指示という表現を用いたが、前述したように、BCに記録された情報は改竄不可能であることから取引の履歴はBCが存在する限り追跡可能・検証可能である。
- 以下の1行目は、BCネットワークに対して、追跡管理番号B000のこれまでの状態を全て表示する指示である。2行目以降がそのシステム応答内容である。履歴は上から順に、新しい取引内容となっている。

```
off_chain_data $ peer chaincode query -C mychannel -n basic -c '{"Args":["GetAssetHistory","B0000"]}'  
[{"record":{"Date":"","ID":"B0000","Type":"","AssetName":"","Owner":"","Price":0,"Momentum":0},"txId":"fe593a0eb0f4d560d5ef654433adbd3dcd7e648678c8e68e4309476b52f3c18f","timestamp":"2022-08-10T23:22:10.558Z","isDelete":true}, {"record":{"Date":"20210101","ID":"B0000","Type":"asset","AssetName":"KLMBonds1","Owner":"KLM","Price":30000000,"Momentum":300000},"txId":"7030521e0bc43ed6f02e0217d0bb22539e152476bd02c4f1adf82ba7447263f1","timestamp":"2022-08-10T23:22:03.081Z","isDelete":false}, {"record":{"Date":"20210101","ID":"B0000","Type":"asset","AssetName":"ShareHoldersBonds1","Owner":"KLM","Price":30000000,"Momentum":300000},"txId":"36bb0759c3cd46e7bf3a9ed984233d6e2355ba11e612d16cc2708c207bac497b","timestamp":"2022-08-10T23:21:59.797Z","isDelete":false}, {"record":{"Date":"20210101","ID":"B0000","Type":"asset","AssetName":"ShareHoldersBonds1","Owner":"ShareHolders","Price":30000000,"Momentum":300000},"txId":"ecba2f77e2cc96c9f32ab5b506c41ea3775a66f08ef19c00daff09f4e78a9af7","timestamp":"2022-08-10T23:21:56.32Z","isDelete":false}]
```

Currentprice	Asset
20240803/B00003/XYZDeposit2/300000/B00004/	20240803/B00003/XYZDeposit2/300000/
20240809/B00007/XYZInventory1/100000/B00005/	20240809/B00007/XYZInventory1/100000
20240810/B00012/XYZDeposit6/16250/B00009/	20240810/B00012/XYZDeposit6/16250/B
20240810/B00010/200000/1/B00008/	20240810/B00010/200000/B00008/
20240810/B00013/183750/B00011/	20240810/B00013/183750/B00011/
<b>currentNetLoss0</b>	<b>Netloss0</b>
<b>currenttotal:800000</b>	<b>total:800000</b>

表示年月日操作 2024/08/10

20240810/B00010/199781/1/B00008/

183750/B00011/

表示年月日操作 2024/08/14

Currentprice	Asset
20240803/B00003/XYZDeposit2/300000/B00004/	20240803/B00003/XYZDeposit2/300000/B00004/
20240809/B00007/XYZInventory1/100000/B00005/	20240809/B00007/XYZInventory1/100000/B00005/
20240810/B00012/XYZDeposit6/16250/B00009/	20240810/B00012/XYZDeposit6/16250/B00009/
20240810/B00010/199781/1/B00008/	20240810/B00010/200000/B00008/
20240810/B00013/191750/B00011/	20240810/B00013/183750/B00011/
<b>currenttotal:807781</b>	<b>total:800000</b>

199781/1/B00008/

16250/B00009/

オレンジの枠で囲んである部分は、表示日時を決定するためのものである。  
 ここで表示日時を決定した上で、ブロックチェーンから引き出したデータを読み込むと、データの作成時点ではなく、データ表示時点として設定した日時における現在の価値を自動的に計算することができる。  
 つまり、簿記・会計情報は、外部データベースや、計算式を提供することによって、作成時点とは異なる時点においてもその時点時点の簿記・会計データを情報利用者に提供することができる。

Currentprice		Currentprice	Loss	Profit	Currentprice
20240803/B00004/300000/1/B00003/		2024-08-14/auto_calc/219/1/2024-08-14/B00010			2024-08-14/auto_calc/8000/50/2024-08-14/B00013
20240801/B00002/500000/1/B00001/					
<b>currentNetIncome:7781</b>		<b>currentNetIncome:7781</b>			
<b>currenttotal:807781</b>		<b>currenttotal:8000</b>	<b>total:0</b>	<b>total:0</b>	<b>currenttotal:8000</b>

- 評価益、減価償却費の発生（ただし実際には記録されていないので、計算による結果としてcurrentpriceの欄に表示するようにしている）も自動的に計算される。

# 検討課題

- 減価償却資産と異なり、金融商品について時価を計算するためには、外部データベースの使用が必要となる場合が多い。
- わが国においては、こうしたデータベース利用の提供は遅れており、企業の保有する金融商品によっては、時価の自動取得が困難である場合も考えられる。
- 現状において、環境面で課題は残るものの、ブロックチェーン時代における簿記システムにおいては、全てをブロックチェーン上で処理するのではなく、ブロックチェーンの特徴を活かしながらも、企業外部のネットワークとも連携しながら、入力した情報が表示時点によって逐次アップデートされ、情報利用者に提供されることが望ましいと考える。

# 参考文献

- ・ BLOCKDATA, 2021, 「BLOCKCHAIN ADOPTION BY THE WORLDS TOP 100 PUBLIC COMPANIES(REPORT)」,BLOCKDATA.
- ・ 石川純治,1994,『経営情報と簿記システム』,森山書店
- ・ 片野一郎. 1975. 『簿記の手ほどき（改訂版）』同文館.