# IOT データベースブロックチェーン

## IOT DATA BLOCK CHAIN

モノのインタネットデータ生態体系を構築する

# 目次

# 第一章　IOT データベースブロックチェーン･万事データ連携

## IOT データベースブロックチェーン(IOT Database Blockchain)･万事インターネット(Internet of Everything)

データベースブッロクチェーンはモノのインタネット理論とブッロクチェーン技術を融合して、万事インターネットのビッグデータフォームを作ることとなります。伝統的なモノのインタネットフォームは中心化技術を採用しており、データの収集や運用ではいずれもモノのインタネットの参入と前提となっています。信用体制の発展が遅くなり、モノのインタネット市場の発展にも大きな支障が出て来ました。モノのインタネットの途中で出てきた問題に対して、データベースブロックチェーンはブロックチェーン技術の応用はモノのインタネット業界はデータの引き取り、データの開発、データの共同存在の問題があり、ブロックチェーンーモノのインタネットービッグデータを統合して、新たなもののインタネット体制を構築します。

データベースブロックチェーンはモノのインタネット関与するすべての工程でのデータを保存して、資産化の転化に実現します。同時に、データベースブロックチェーンはブロックチェーンの特徴を借りて、基本データベースに信用体制を作ります。データベースブロックチェーンの中では、修正不可性と検索可能性があり、関係者は取引の公正体制に対して絶対的な信用があり、取引の通常運用を保障します。

LDBC（ロジスティックスデータ　ブロック　チェーン）はデータベースブロックチェーンの基礎対象であります。LDBC はデータベースブロックチェーンを通じて、モノのインタネットの中の商品流通データを資産化にすることを促成します。伝統的なデータ収集方法を覆し

ました。データベースブロックチェーンはモノの基礎流通データを通して、他のスマート契約を作り出し、最終的に高使用性のあるもののインタネットのビッグデータを促成します。

# 第二章 IOT データベースブロックチェーン(IOT Database Blockchain) テクノロジー

## 1.1 IOT データベースブロックチェーン

IOT データベースブロックチェーンの目的は、オン・チェーン・メタ・プロトコル、スクリプト、スマート・コントラクト、クロスチェーン・コラボレーション、分散ストレージのコンセプトに基づいてテクノロジを統合し、強化することで、開発者はデータ・プレゼンテーションとデータ・ストレージ・プラットフォームを作成できます。 スケーラブルで、標準化され、完全に機能し、開発しやすく、コラボレーションしやすい任意のコンセンサスベースです。 Turing complete なプログラミング言語を内蔵した究極の抽象基本層を作成することで、誰でも契約を作成し、アプリケーションを分散させることができ、自由に定義された所有権ルール、取引方法、状態遷移機能を設定できます。

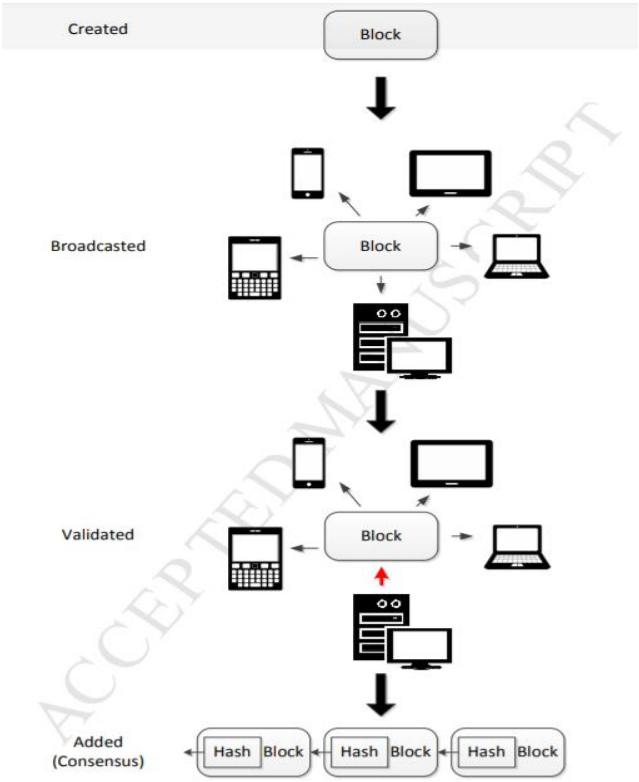


図 1 典型的なブロックチェーンワークフロー

1.2 IOT データベースブロックチェーンアカウント

アカウントシステムでは、状態は「アカウント」（各アカウントは 20 バイトのアドレス）というオブジェクトと 2 つのアカウント間で価値と情報を転送する状態で構成されます。LDBC アカウントには 4 つの部分：

•乱数、トランザクションごとに 1 回しか処理できないことを判断するために使用されるカウンタ

•現在の勘定残高

•契約コード

•アカウントの保存（デフォルトは空です）

LDBC は、IOT データベースブロックチェーン内の主な暗号化燃料であり、取引コス

トを支払うために使用されます。 一般に、2 つのタイプのアカウントがあります：すべての外部アカウント（秘密鍵によって制御される）と契約アカウント（契約コードによって制御される）。すべての外部アカウントにコードはなく、ユーザーはトランザクションを作成して署名することで、外部アカウントからメッセージを送信できます。 契約アカウントがメッセージを受け取るたびに、契約内のコードが有効になり、内部ストアの読み書きや他のメッセージの送信や契約の作成が可能になります。

## 2.1 1.3 ニュースと取引

まず、外部エンティティまたは契約によってメッセージを作成することができますが、ビットコイントランザクションは外部でのみ作成することができます。 第 2 に、メッセージはオプションでデータを含むことができます。 第 3 に、メッセージの受信者が契約アカウントである場合、応答することを選択することができます。つまり、IOT データベースブロックチェンジには関数の概念も含まれます。

IOT データベースブロックの「トランザクション」は、外部アカウントから送信されたメッセージを格納する署名パケットを指します。トランザクションには、メッセージの受信者、送信者を確認するために使用される署名、通貨口座の残高、送信予定のデータ、STARTGAS および GASPRICE という 2 つの値が含まれます。指数関数的な爆発とコードの無限ループを防ぐために、各トランザクションは、実行時に発生する初期メッセージやすべてのメッセージを含め、実行コードによって発生する計算ステップを制限する必要があります。 STARTGAS は制限事項であり、GASPRICE は計算ステップごとに鉱山者に支払うために必要なコストです。実行中の取引中に「燃料が使い果たされた」場合、すべてのステータスの変更は元の状態に戻りますが、すでに支払った取引手数料は回収できませ

ん。実行中の取引が中断されたときに燃料がまだ残っている場合、燃料は送付者に返されます。作成契約には、個別の取引タイプとそれに対応するメッセージタイプがあります。契約のアドレスは、アカウントの乱数と取引データのハッシュに基づいて計算されます。

メッセージメカニズムの重要な結果は、IOT データベースブロックの「プライマリシチズン」プロパティです。契約は、メッセージを送信したり他の契約を作成する権利を含め、外部アカウントと同じ権限を持ちます。これにより、契約は複数の異なる役割を同時に果たすことができます。例えば、ユーザは、分権化された組織（契約）のメンバーを、仲介アカウント（別の契約）にすることができ、カスタマイズされた量子ベースのランバート署名（第 3 の契約）を使用する偏心した個人と、5 つの秘密鍵で保護されたアカウント（4 番目の契約）。IOT データベースブロックは、契約の各当事者がどのような種類のアカウントを扱っているかを気にする必要はありません。

## 2.2 1.4 IOT データベースブロックチェーンの状態遷移関数

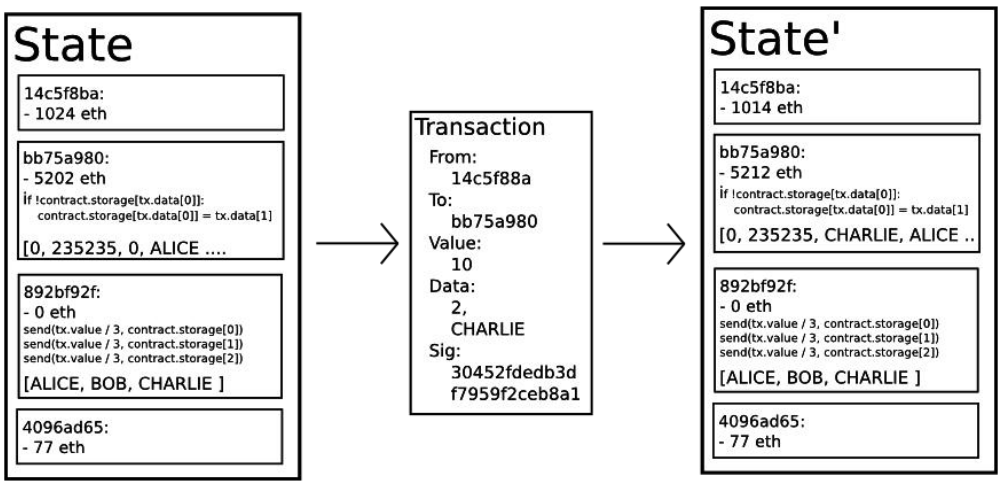


図 2 状態遷移関数

IOT データベースブロックチェーンの状態遷移関数：APPLY（S、TX） - > S 'は、以下のように定義できます：

1 トランザクションの形式が正しいか（正しい値であるか）署名が有効であるか、乱数が送信者のアカウントの乱数と一致するかをチェックします。そうでない場合、エラーが返されます。

2 計算トランザクション手数料：手数料= STARTGAS * GASPRICE、署名から送付者の住所を決定する。送信者のアカウントから取引手数料を差し引き、送信者の乱数を増やします。口座残高が不足している場合は、エラーが返されます。

3 初期値 GAS = STARTGAS を設定し、トランザクション内のバイトに応じて一定量の燃料値を減算します。

4 送信者のアカウントから受信者のアカウントに値を転送します。受信アカウントがまだ存在しない場合は、アカウントを作成します。受領口座が契約の場合は、コードまたは燃料がなくなるまで契約コードを実行します。

5 送金者の口座に十分な金額がないか、コードの実行に燃料がなくなった場合、振り替えが失敗し、元の状態に戻ります。取引手数料は引き続き支払う必要があります。取引手数料は鉱山者口座に追加します。

6 そうでなければ、残りの燃料は全て送付者に返却され、消費された燃料は取引コストとして鉱夫に送られる。

たとえば、契約のコードが次のようになっているとします：

if ! contract.storage [msg.data [0]] :

contract.storage [msg.data [0]] = msg.data [1]

実際、注意すべきことがあり、現実の契約コードは基本的な仮想マシンコードで書かれています。 契約メモリは、最初は空であり、価格は 10 コインと 2000 燃料であり、燃料価格は 0.001 コインであると仮定する。 そして、2 つのデータフィールド値が[2、'CHARLIE] [3]であるトランザクションの後、状態遷移関数の処理は以下のようになる：

1 トランザクションが有効で、フォーマットが正しいかどうかを確認します。

2 取引の送信者に少なくとも 2000 * 0.001 = 2 コインがあることを確認します。その場合は、送信者のアカウントから 2 コインを引きます。

3 最初にガス= 2000 を設定する。トランザクションの長さは 170 バイト、バイトあたりのコストは 5 である。その中から 850 を引いた値であるため、残りのバイト数は 1150 です。

4 送信者アカウントから 10 コインを差し引き、10 コインを契約アカウントに追加します。

5 コードを実行します。 この契約では、コードを実行するのが簡単です。契約メモリのインデックス 2 が使用されているかどうかを確認します。 使用されていない場合は、その値を CHARLIE に設定します。 これが 187 単位の燃料を消費すると仮定すると、残りの燃料は 1150-187 = 963 である。

6 963 * 0.001 = 0.963 コインを送信者の口座に追加して、最終ステータスに戻ります。

トランザクションを受け取る契約がない場合、すべてのトランザクションコストはGASPRICE にトランザクションバイトの長さを掛けたものに等しく、トランザクションデータはトランザクションコストとは無関係です。さらに、契約によって開始されたメッセージは、発生する計算に燃料制限を割り当てることができることに留意すべきである。サブ計算された燃料がなくなると、メッセージが送信された時点の状態に戻るだけです。したがって、トランザクションのように、契約では、生成するサブ計算に厳しい制限を設定することによって、コンピューティングリソースを保護することもできます。

## 2.3 1.5 コードの実行

IOT データベースブロックコントラクトのコードは、「IOT データベースブロック仮想マシンコード」と呼ばれる低レベルのスタックベースのバイトコード言語で書かれています。コードは一連のバイトで構成され、各バイトは一種の操作を表します。一般に、コードの実行は無限ループです。プログラムカウンタが 1 増加すると（初期値はゼロ）、コード実行が完了するか、エラー、STOP または RETURN 命令が発生するまで、演算が 1 回実行されます。この操作では、データを格納するための 3 種類の領域にアクセスできます。

●スタックは、先入れ先出し形式のデータストアで、32 バイトの値をスタックにプッシュまたはプッシュアウトできます。

●メモリ、無限に拡張可能なバイトキュー。

●契約の長期保存、秘密鍵と秘密鍵の値の保管、秘密鍵と秘密鍵の値はすべて 32 バイトです。計算終了時にリセットされるスタックやメモリとは異なり、格納されたコンテンツは長期間保存されます。

コードは、ブロックチェーンデータにアクセスするのと同じように、受信メッセージの値、送

信者、およびデータにアクセスできます。コードは、データのバイトキューを出力として返すこともできます。

仮想マシンコードの正式な実行モデルは、驚くほど簡単です。IOT データベースブロック仮想マシンが実行されている場合、その完全な計算状態はタプル（block_state、トランザクション、メッセージ、コード、メモリー、スタック、pc、ガス）によって定義できます.block_state はすべてのアカウントの残高と記憶域を含まっています。実行の各ラウンド中に、現在の命令はコードの pc（プログラムカウンタ）バイトを呼び出すことによって検出され、各命令はタプル自体にどのように影響するかを定義します。たとえば、ADD は 2 つの要素をポップし、それらの合計をスタックに挿入し、ガス（燃料）を 1 つ引いて pc を 1 つ追加します。SSTORE は上位 2 つの要素をポップし、2 番目の要素を各要素で定義された最初のコントラクト格納場所に挿入し、ガス値を 200 まで減算し、pc に 1 を加算します。ジャストインタイムコンパイルによって最適化する方法はたくさんありますが、IOT データベースブロックの基本的な実装は数百行のコードで実装できます。

## 2.4 1.6 ブロックチェーンとマイニング

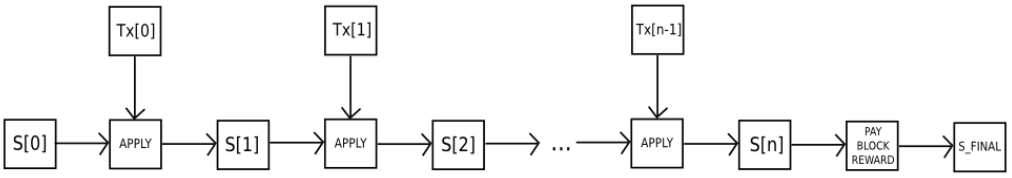


図 3 ブロックチェーンと鉱山の地図

いくつかの違いがありますが、IOT データベースブロックのブロックチェーンは、多くの面で

Bitcoin のブロックチェーンに似ています。ブロックチェーンアーキテクチャの違いは、IOT データベースブロックのブロックにトランザクションレコードと最近のステータスだけでなく、ブロック番号と難易度の値も含まれていることです。 IOT データベースブロックのブロック検証アルゴリズムは次のとおりです：

1 ブロックによって参照される前のブロックが存在し、有効であるかどうかを確認します。

2 ブロックのタイムスタンプが前の参照ブロックよりも大きく、2 分未満であることを確認します。

3 ブロック番号、難易度の値、トランザクションのルーツ、ターロール、燃料制限（Ethereum 固有の概念の多く）が有効であるかどうかを確認します。

4 ブロックのワークロード認証が有効かどうかを確認します。

5 前のブロックの STATE_ROOT に S [0]を割り当てます。

6 ブロックのトランザクションリストに TX を割り当てます。 合計で n 回の取引があります。 0 ... n-1 に属する i に対して、状態遷移 S [i + 1] = APPLY（S [i]、TX [i]）が実行される。 変換エラーが発生した場合、またはここでプログラムを実行するために使用されたガスが GASLIMIT を超える場合、エラーが返されます。

7 S [n]を使用して S_FINAL を割り当て、ブロックボーナスをマイナーに支払う。

8 S-FINAL が STATE_ROOT と同じであることを確認します。 同じ場合、ブロックは有効です。それ以外の場合、ブロックは無効です。

IOT データベースブロックの確認効率は、Bitcoin をはるかに超えています。その理由は、状態がツリー構造に格納され、追加のブロックごとにツリー構造のごく一部を変更す

る必要があるからです。したがって、一般に、2 つの隣接するブロックのツリー構造の大部分は同じであるはずです。データを一度格納することはポインタ（すなわち、サブツリーハッシュ）を使用して 2 回参照することができる。 "Patricia Tree"というツリー構造は、ノードの変更だけでなく、ノードの挿入と削除も可能な Merkel ツリーの変更を含む、これを実現します。

一般に、IOT データベースブロックチェーンの完璧な例は、物流情報を共有するという問題を解決するための自己強制的な報酬です。 オリジナルの環状計算により、データ共有が標準化されたコンセンサスになり、ブロックチェーンの認識とコンセンサス報酬が向上します。

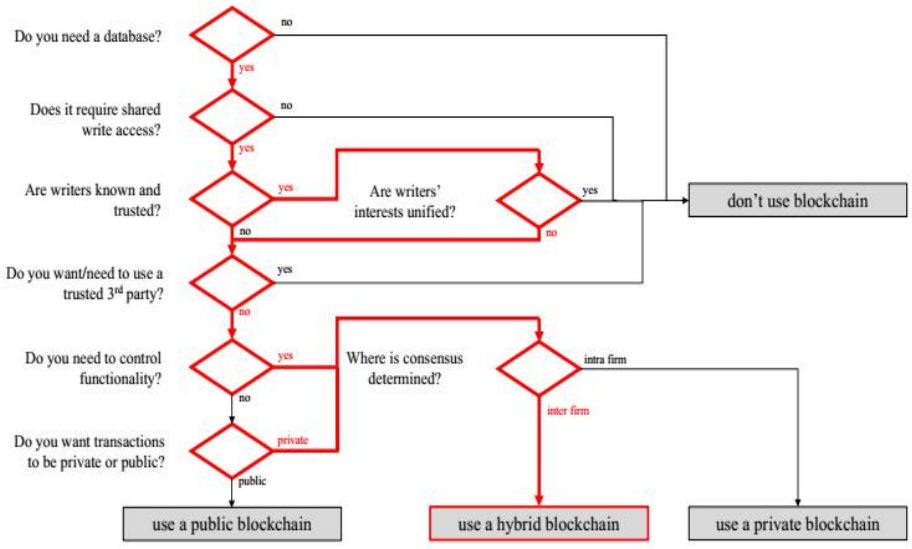


図 4 使用するブロックチェーンの種類

## 2.5 1.7 改良された Ghost プロトコル

ゴーストプロトコルの背後にある動機は、ブロックの高いブロックレートのために現在の

高速認識ブロックチェーンが低セキュリティで悩まされていることです。ブロックがネットワーク全体に広がる一定の時間（t に設定）を費やす必要があるからです。鉱夫 A がブロックを掘った後、鉱夫 B が A のブロックが B に広がる前に別のブロックを掘り起こすと、鉱夫 B のブロックは廃止され、ネットワークセキュリティに寄与しません。さらに、集中化の問題もあります。A が全ネットの 30%の計算能力を持ち、B が 10%の電力を持つ鉱山である場合、A は時間の 70%が無効ブロックを生成し、B は時間の 90%が無効なブロックを生成するリスクに直面しています。したがって、失効率が高い場合、A は単にコンピューティングパワーの高いシェアのために効率的になります。これらの 2 つの要素を組み合わせることで、迅速に生成されたブロックチェーンによって、実際にマイニングを制御できるコンピューティングパワーシェアを持つ鉱山プールが得られる可能性が高くなります。プロセスのパワーシェア。どのチェーンが「最長」であるかを計算する際に廃棄ブロックを含めることで、Ghost プロトコルはネットワークセキュリティの低下の最初の問題を解決します。すなわち、親ブロックおよび先祖の祖先ブロックに加えて、祖先ブロックの無効化後の子孫ブロックも、それをサポートする最大ワークロード確認を有するブロックを計算するために追加される。 IOT データベースブロックは、新しいブロック確認に貢献する廃棄ブロックの報酬の 87.5%を「叔父のブロック」として支払う。計算に含まれる「甥ブロック」には、報酬の 12.5%が贈られます。しかし、取引手数料は叔父のブロックには与えられません。

IOT データベースブロックは、単純なバージョンのゴーストプロトコルを実装しており、これは第 5 レベルにしかなりません。廃棄ブロックは、親のブロックの第 2 世代から第 5 世代の子ブロックまで、より離れた子孫ブロック（第 6 世代の親ブロックまたは第 3 世代祖父のブロックの子孫ブロック）。これにはいくつかの理由があります。第 1 に、無条件のゴーストプロトコルは、与えられたブロックの叔父ブロックが合法である計算に余分な複雑さを与

える。第二に、IOT によって使用される補償を伴う無条件のゴーストプロトコルは、攻撃者の鎖ではなく連鎖上の鉱山鉱山者に鉱山権を奪う。最後に、ブロックアウト時間が 15 秒であってもインセンティブを持つ 5 レベルのゴーストプロトコルが 95%以上の効率を達成する一方、中央集中からの 25%の電力消費者の利益は 3%未満であることが示されています。

ブロックチェーンに投稿された各トランザクションはダウンロードと検証のコストを消費するため、スパミングトランザクションを防止するための取引手数料を含む規制メカニズムが必要です。 Bitcoin が使用するデフォルトの方法は、ゲートキーパーとしての鉱夫に依存し、ダイナミックな最低料金を設定する純粋に任意の取引手数料です。この方法は「市場ベース」であり、鉱夫や取引の送信者が需要と供給に基づいて価格を決定できるため、この方法は Bitcoin コミュニティで首尾よく受け入れられます。しかし、このロジックの問題は、トランザクション処理が市場ではないことです。トランザクション処理を鉱山者が提供するサービスとして送信者に直感的に解釈することは興味深いが、実際には、鉱夫の取引は各ノードの処理を必要とするため、トランザクション処理コストの最大部分は第三者が負担する取引を含めるかどうかを決める鉱夫よりも。その結果、建設現場の悲劇が起こりやすい。

しかし、特殊で簡潔ではない単純な仮定が与えられたとき、この市場ベースのメカニズムの抜け穴は、その影響を奇跡的に排除する。 引数は次のとおりです。 前提条件：

1 貿易は k ステップをもたらし、貿易を含む任意の鉱夫に報酬 kR を提供する。ここで、R は取引発行者によって設定され、k と R は（事前に）鉱山者に（ほぼ）可視である。

2 各ノードが各ステップを処理するコストは C です（つまり、すべてのノードの効率は

一貫しています）。

3 それぞれに同じコンピューティングパワー（つまり、ネットワーク全体のパワーの 1 / N）を持つ N 個のマイニングノードがあります。

4 マイニングなしの完全ノードはありません。

予想される報酬がコストを上回っている場合、鉱夫は喜んで鉱山する。このようにして、鉱夫は次のブロックを処理する 1 / N 機会があるので、期待利益は kR / N であり、鉱夫の処理コストは単純に kC です。 kR / N> kC、すなわち R> NC の場合、鉱夫は取引を含むことを望んでいる。 R はトランザクションの送信者によって提供されるステップあたりのコストであり、これは、取引を処理することにより鉱夫が利益を受ける下限であることに留意されたい。 NC は、ネットワーク全体で操作を処理するコストです。したがって、鉱山者は、コスト以上の利益をもたらす取引を含める動機があるだけです。

しかし、これらの前提および実際の状況からいくつかの重要な逸脱があります：

1.追加検証時間はブロックのブロードキャストを遅延させ、ブロックが廃棄ブロックになる可能性を高めるので、トランザクションを処理する鉱夫は他の検証ノードよりも高いコストを支払う。

2.マイニングなしの完全ノードがあります。

3.実際のコンピューティングパワーの分布は極端に不均等になる可能性があります。

4.インターネットを破壊しようとする真の投機家、政治家、狂人がいる。 彼らは、他の検証ノードよりもはるかにコストが安くなるように、巧みに契約を設定することができます。

上記の最初の点は、より少ないトランザクションを含むように鉱夫を駆動し、2 番目の

点は NC を増加させます。したがって、これらの 2 つのポイントの影響は、少なくとも部分的に相殺されます。 3 番目と 4 番目のポイントは主な問題です。解決方案として我々は浮動の上限を簡単に設立しました。BLK_LIMIT_FACTOR 回の長期指数移動平均より多くのオペランドを含むブロックはありません。 具体的には：

blk.oplimit = floor （（ blk.parent.oplimit * （ EMAFACTOR-1 ） + floor（parent.opcount * BLK_LIMIT_FACTOR））/ EMA_FACTOR）

BLK_LIMIT_FACTOR と EMA_FACTOR は、一時的に 65536 と 1.5 に設定された定数ですが、後で解析して調整することができます。

## 2.6 1.8 計算そして Turing Complete

IOT データベースブロックマシンは Turing Complete であることを強調する必要があります。 仮想マシンコードが無限ループを含む想像できる計算を実装できることを意味します。 IOT データベースブロック仮想マシンコードでループを実装する方法は 2 つあります。まず、JUMP 命令は、プログラムをどこかに戻すことができます。また、x <27：x = x * 2 のような条件文が条件付きジャンプを実行できるようにする JUMPI 命令。 第 2 に、契約は他の契約を呼び出すことができ、再帰によってループを達成する可能性があります。当然のことながら、悪意のあるユーザーは、鉱夫やすべてのノードを無限ループに強制することによって強制的にシャットダウンすることができますか？ この問題は、コンピュータサイエンスの停止問題のために発生します。一般的な意味で、特定のプログラムが限られた時間内に実行を終了できるかどうかを知る方法はありません。

私たちのソリューションは、トランザクションごとに実行される計算されたステップの最大数を設定することによって問題を解決します。 それを超えた場合、計算は元に戻りますが、

それでも支払う必要があります。 ニュースは同じように機能します。 このシナリオの背後にあるモチベーションを示すには、次の例を考えてみてください：

攻撃者は無限ループを実行する契約を作成し、ループをアクティブにするトランザクションを鉱山業者に送信します。トランザクションを処理し、燃料がなくなるまで無限ループを実行します。 燃料が使い果たされて取引が途中で停止しても、取引はまだ正しい（元の位置に戻る）と、鉱夫は依然として攻撃者からの各ステップのコストを稼ぐ。

攻撃者は非常に長い無限ループ意図を作成して、鉱夫が長時間計算するようにします。これにより、計算が終了する前にいくつかのブロックが生成され、鉱夫はトランザクションを収集して手数料を得ることができません。 しかし、攻撃者は STARTGAS 値を発行して実行可能ステップの数を制限する必要があるため、計算機にはあまりにも多くのステップがかかることが事前にわかります。

攻撃者は、send（A、contract.storage [A]）のような契約を含む契約を見る。 contract.storage [A] = 0 とし、最初のステップを実行するのに十分なだけのトランザクションを送信しますが、2 番目のステップは実行しません。 トランザクション（つまり、勘定残高を削減することなく払い戻し）。 契約の作成者は、実行が途中で停止するとすべての変更が元に戻されるため、同様の攻撃の防御について心配する必要はありません。

金融契約は、リスクを最小限に抑えるために 9 つの私的データ発行者の中央値を抽出することによって機能します。 攻撃者はデータプロバイダの 1 つを引き継ぎ、可変アドレス呼び出しメカニズムを可変データプロバイダとして設計して無限ループを実行し、燃料がなくなるのを待ってこの契約の資金を要求するように説得します。 しかし、このような問題を防ぐために、契約書には燃料制限を設定することができます。

Turing complete の代替案は Turing incomplete チューリングが完了すると、JUMP 命令と JUMPI 命令が存在しない Turing incomplete が置換されます。また、各契約では、特定の時間に呼び出しスタックにコピーが 1 つだけ存在することが許可されます。このようなシステムでは、契約の実行コストはその規模によって決まるため、前述の料金体系と私たちを取り巻く解決策の不確実性は必要ないかもしれません。さらに、Turing incomplete は大きな制限でもありませんが。これまで想定していたすべての契約例では、1 つのサイクルだけを循環させる必要があり、このサイクルでも 26 の単一行コードセグメントの繰り返しに置き換えることができます。重大な問題と限られた利益を考慮に入れて Turing complete、なぜ Turing incomplete 言語を使用しないのですか？？ 事実、Turing incomplete さは簡潔な解からは遠いです。どうして？ 以下の契約を考慮してください：

C0：コール（C1）;コール（C1）；

C1：コール（C2）;コール（C2）；

C2：コール（C3）;コール（C3）；

…

C49：コール（C50）;コール（C50）；

C50：（プログラムの 1 ステップを実行してストレージの変更を記録する）

今、このようなトランザクションを A に送信します。私たちは、51 トランザクションで 250 ステップかかる契約をしています。鉱夫は、契約ごとに最大の実行可能ステップを維持しようとするかもしれないし、再発を呼び出す契約の可能な実行ステップを計算して、そ

のような論理爆弾を事前に検出するかもしれないが、これは鉱夫が他の契約を作ること禁止することになります。（なぜなら、上記の 26 の契約の作成と実行は簡単に 1 つの契約に入れられるからです）。別の問題は、メッセージのアドレスフィールドが変数であるため、一般に、契約が呼び出す他の契約を事前に知ることさえできないことがあります。このように、私たちは最終的に驚くべき結論を得ました.Turing complete の管理は驚くほど簡単ですが、同じコントロールがない場合、Turing incomplete さの管理は驚くほど困難です- プロトコル Turing complete を使用しない理由?

## 2.7 1.9 鉱業の地方分権化

IOT データベースブロックの目的は、専用ハードウェアの利点を取り除くために、十分に広い計算領域を持つ 1000 個の乱数ごとに一意のハッシュをランダムに生成する関数に基づくマイニングアルゴリズムを使用することです。個々のユーザーはそれぞれ専用のラップトップまたはデスクトップマシンを使用して、ほぼ一定量の鉱業活動を無料で行うことができますが、100%の CPU 使用後に電力やハードウェアの費用を支払う必要があります。ASIC 鉱業会社は、最初のハッシュから電力とハードウェアを購入する必要があります。したがって、集中収益を（E + H）/ E 以下に抑えることができれば、ASIC を製造しても通常の鉱夫にはまだ生き残る余地があります。さらに、ブロックチェーン全体へのアクセスを必要とするマイニングアルゴリズムを設計し、鉱夫に完成したブロックチェーンを格納させたり、少なくとも各トランザクションを検証できるようにします。これにより、集中型鉱山プールが不要になります。鉱山プールは依然として所得分配を平滑化する際にランダムな役割を果たすことができますが、この機能は集中管理なしに P2P プールでも同様に実行できます。このように、通常のユーザーのほとんどが依然として軽いクライアントを選択することを好みますが、ネットワーク内のフルノードの数を増やすことは、集中化を防ぐのにも役立ちます。

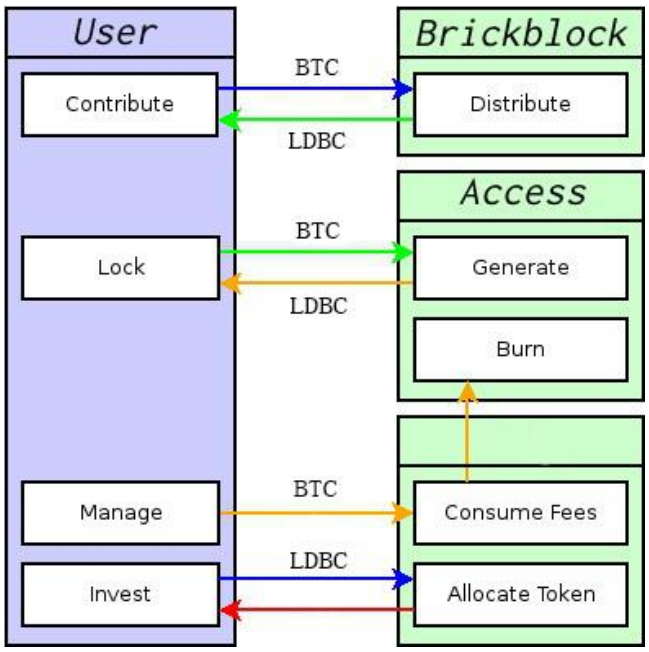


図 5 トークンの種類とその相互作用

## 2.8 1.10 拡張性

拡張性はしばしば注目の焦点です。ビットクロックと同様に、IOT DATABASE CLOCKCHAIN はまた、各トランザクションがこのジレンマに対処するためにネットワーク内のすべてのノードを必要とするという事実に苦しんでいます。Bitcoin の現在のブロックチェーンサイズは約 20GB で、1 時間あたり 1MB の速度で成長します。Bitcoin ネットワークが Visa クラスの 2000tps トランザクションを処理する場合、3 秒ごとに 1MB で増加します（1GB /時、8TB /年）。単純な通貨としての Bitcoin ではなく、チェーンの上に多くのアプリケーションが存在するため、IOT データベースブロックは同様のパターンまたはさらに悪い成長パターンを経験することがあります。しかし、チェーン内のノードが完全にブロックチェーンの履歴を保存するのではなく、ストアの状態だけを必要とするという事実は状況を改善しました。

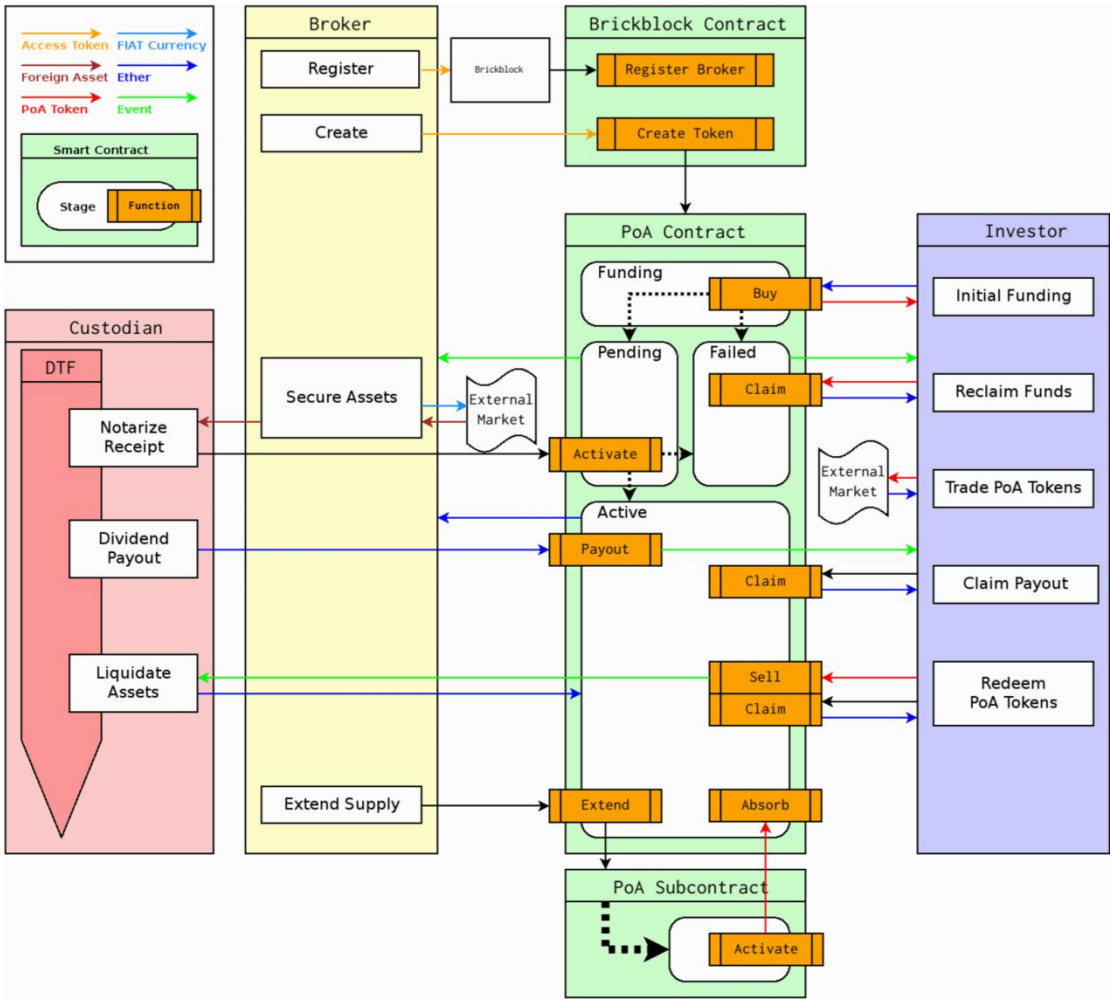


图 6 トークンの周期

大規模なブロックチェーンは、集中化のリスクに直面しています。ブロックチェーンのサイズが 100TB に増加する場合、通常のユーザーは軽い SPV ノードを使用していますが、非常に少数の大規模な商人がフルノードを実行するだけです。完全なノードパートナーシップにおける詐欺のリスクについての懸念が生じます（例えば、ブロック報酬を変更して BTC を与えるなど）。軽いノードは、この詐欺をすぐに検出する方法がありません。もちろん、少なくとも 1 つの正直な完全ノードが存在する可能性があり、数時間後に Reddit のようなチャネルを介して不正な情報が漏洩する可能性がありますが、遅すぎます。ユーザーは既に生成されたブロックを廃止しようと努力しているにもかかわらず、成功した 51%の攻撃の開始と同じ規模の膨大な実行不可能な調整問題に遭遇します。 IOT データベ

ースブロックは、この問題に対処するための 2 つの追加戦略を使用します。第 1 に、ブロックチェーンベースのマイニングアルゴリズムにより、少なくとも各マイナーは完全なノードになることが強制され、ある程度の完全なノードが保証されます。次に重要なことは、各トランザクションを処理した後、中間状態ツリーのルートをブロックチェーンに含めることです。ブロック検証が集中化されていても、正当な検証ノードが存在する限り、検証プロトコルによって集中化を回避することができます。マイナーが不正なブロックを発行した場合、そのブロックは間違ったフォーマットか、状態 S [n]が間違っています。 S [0]は正しいので、最初のエラー状態 S [i]が存在しなければならないが、S [i-1]は正しい。検証ノードは、インデックス i を、パトリシアツリーノードのサブセットとともに提供する。 APPLY（S [i-1]、TX [i]）→S [i]の処理を行う。これらのノードは、結果の S [i]が以前に提供された値と一致するかどうかを見るために計算を実行することが義務づけられます。さらに、悪意のある鉱夫が攻撃のために不完全なブロックを発行し、そのブロックが正しいかどうかを判断するための情報が不十分になることはより複雑です。解決策は質問応答プロトコルであり、検証ノードは取引インデックスを対象とし、情報を受け取るライトノードは、別の鉱夫または検証者がパトリシアノードのサブセットを正しい証拠として提供するまで、対応するブロックを信頼しない。

## 2.9 1.11 レビュー

上記の分散アプリケーションのための契約メカニズムは、ネットワーク全体のアクセス可能な状態を「ハードディスク」として変更できる仮想マシン上のグローバルなネットワークコンセンサスを使用して、任意の人がコマンドラインアプリケーション（基本的には話す）を確立することを可能にします。しかし、ほとんどの人にとって、トランザクション配信メカニズムとして使用されるコマンドラインインターフェイスの適切な使いやすさの欠如は、分散化を魅力

的な選択肢にしています。最後に、完全な「分散アプリケーション」には、基礎となるビジネスロジックコンポーネントと上位のグラフィカルユーザーインターフェイスコンポーネントが含まれている必要があります。 IOT データベースブロッククライアントはウェブブラウザとして設計されていますが、 "LDBC" Javascript API のサポートが含まれています。これは、クライアントがチェーンと対話するのに見られる特定の Web ページによって使用されます。ブロックチェーンや他の分散型プロトコルは、ユーザーが開始した要求を処理するためにサーバーを完全に置き換えるため、「従来の」Web ページの観点からは、これらの Web ページは完全に静的なコンテンツを保持しています。最後に、分散型プロトコルは、LDBC を使用して Web ページを何らかの形で保存することを期待しています。

## 2.10 1.12 IOT データベースブロックの分布ルール

IOT データベースブロックネットワークには、独自の IOT DATABASE 通貨が含まれています。 LDBC は、さまざまなデータ取引に大きな流動性を提供する二重の役割を果たします。さらに重要なことは、取引コストを支払う仕組みを提供することです。

分布モデルは次のとおりです：

●営業活動、コンピューティング・パワー・アロケーション、データ共有を通じ、IOT データベースブロックは 10 年間でおよそ 90 億の LDBC を生み出し、55 億の LDBC を事前に掘削し、開発者と研究者の賃金と報酬の支払いに充当しました。 生態系に投資するプロジェクト。

●27 億 LDBC は、プロジェクト開発および私募の早期寄稿者に配分される。

●2019 年にリングコンピューティング標準がオンラインになった後にリリースされるデータ寄付賞として、23 億 LDBC が財団会計に含まれる予定です。

●5 億人の LDBC が、開発者やコミュニティの初期建設を基礎として報酬を与えるために使用される。

●3 億 5 千万の LDBC は、オンラインになってから毎年採掘されており、10 年間で合計 35 億の LDBC が掘られている。

永久線形成長モデルを解体し分解することで、Bitcoin の富が過度に集中するリスクが軽減され、IOT データベースの通貨を取得して保持するインセンティブを維持しながら、現在および将来の人々がお金を獲得する公正な機会が得られます。 長期的には、「マネーサプライの成長率」はゼロになる傾向があるからです。 時間が経つにつれて、不注意と死亡による通貨の損失が常に起こります。 通貨の損失が毎年のマネーサプライの固定比率であると仮定すると、最終的な総通貨は、年間通貨を年間通貨で割った値に等しい値で安定する（例えば、供給が 11 倍に達したとき 0.11x が発掘される 毎年 0.11x が失われ、平衡に達する）。リニア発行方式に加えて、Bitcoin が好きな LDBC の供給の伸び率は、長期的にはゼロになる傾向があります。

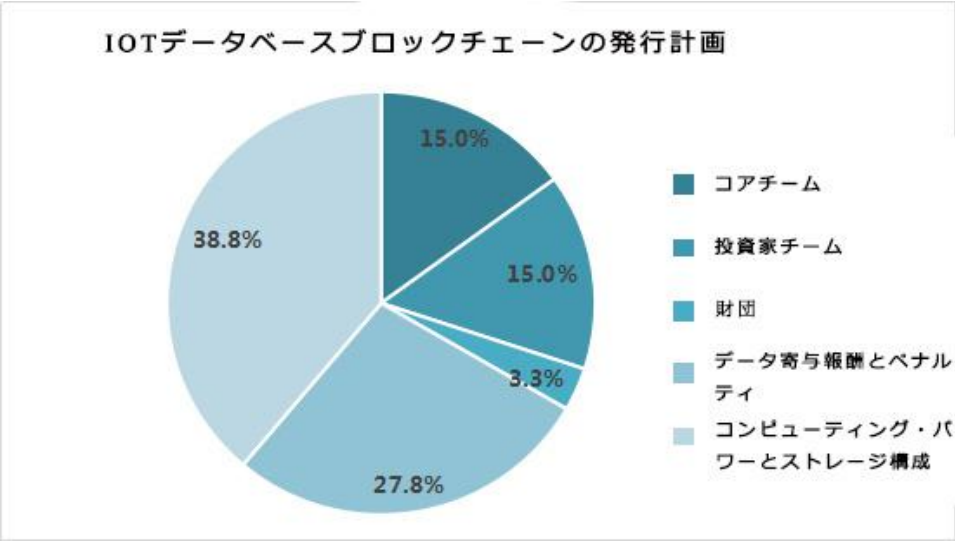


図 7 IOT データベースブロックチェーンの課題

トークンの数：90 億

15% - American Cloud Block chain Network Technology Co.、Ltd.、アジア太平洋地域の IOT データベースブロックチェーンコアチーム

15% - 投資家チーム

8.9% - 財団

27.8% - データ寄与報酬 20%

38.9% - コンピューティングパワーとストレージの構成

2.11 1.13 結論

IOT データベースブロックプロトコルは、非集中型ストレージ、分散型コンピューティング、プロトコル、数多くの同様のコンセプトによって確立された分散型アプリケーションを共有するために、分散型データを中心としています。 物流業界の効率性を根本的に改善し、経済層を追加することで他の P2P プロトコルを強力にサポートする可能性があります。 結局のところ、お金とは関係のない多数のアプリケーションが存在します。

IOT データベースブロックプロトコルによって実装される任意の状態遷移の概念は、独自の可能性を持つプラットフォームを提供します。 データストレージやファイナンスなどの単一目的向けに設計されたクローズドプロトコルとは異なり、オープンエンドで設計されています。 我々は、基盤層として、来るべき数多くの物流業界や非産業協定を実現することが非常に適していると考えています。

# 第三章　チームメンバーご紹介

| | |
|---|---|
| Yanomiyabi | 環形計算発明者、ブロックチェーン応用専門家、多数の日本取引所の基幹開発に参加しました。 |
| Stephen Temple | マサチューセッツ工科大学の天才少年、多数のブロックチェーン特許獲得者です。 |
| Cheong Ben | シンガポールのもののインタネット応用専門家、日本国籍で IOT データベースブロックチェーン理論の先駆者で、シンガポール科学設計大学の博士です。 |
| Haitao Feng | 原百度ブロックチェーンの基幹開発の技術専門家、長年の基幹開発経験があります。 |
| Xiaoming Peng | 原産貿送の CEO で、物流データの専門家です。 |

# 第四章 顧問

| | |
|---|---|
| Rachel Wilson | 教授、データ安全専門家でビットコイン コミュニティの早期メンバーです。 |
| Michael Graetzel | 教授、光とナノメートル技術業界のリーダレベルの人です。 |
| Shinichi Mochizuki | 教授、日本京都大学の教授で数学家、遠アベル幾何学のリーダレベルの人です。 |

# 第五章 免責事項

この文書は、このアイテムの販売および購入に関する関連する意見を構成しない情報を伝達するためのものです。 上記の情報または分析は、投資判断を構成するものではありません。この文書は、投資に関するアドバイス、投資意図、または教育投資を構成するものではありません。

この書類は、有価証券、契約書またはコミットメントを形式で提供する行為を構成するものではありません。

関係のあるユーザーは、このプロジェクトのリスクを明確に理解しています。 投資家が投資に参加すると、プロジェクトのリスクを理解し、それを受け入れ、それに対応するすべての結果を個人的に負担する意思があることを示します。

運営チームは、プロジェクトに参加することによって生じた直接的または間接的な損失を一切負いません。