Factom

ブロックチェーン上の監査証跡によって、守られたセキュアなビジネスプロセス

貢献者: Paul Snow, Brian Deery, Jack Lu, David Johnston, Peter Kirby

アドバイザー: Adam Stradling, Shawn Wilkinson, Jeremy Kandah, Dexx, Marv Schneider, Steven

Sprague, Andrew Yashchuk

レビュワー: Vitalik Buterin, Luke Dashjr, Ed Eykholt, Ryan Singer, Ron Gross, J.R. Willett, Dustin Byington

2014年11月17日　第一版

要旨

“Honesty is subversive” – Paul Show

現在のグローバル経済では表象化された信頼が圧倒的に不足しています。信頼が不足していることによって、大量の調査と、データの検証に大量の資源が注ぎ込まれています。そのため、現在の効率性と、投資対効果、繁栄を妨げています。
更に言えば、2010年のアメリカのリーマン・ショックのような事故を考えると、非効率なだけでなく、失敗しやすい、ひどく不正確なプロセスを取っている事が分かります。Factomでは、正確で、検証可能な、改変不可能な、調査の跡を世界に供給することで、盲目的な信頼をしなければならない必要をなくします。

昔は、記録を保護することは難しく、同期することも困難で、手作業が加わってしまうために、本当に意味で検証することは不可能でした。コンピューターがこれらのタスクの幾つかを自動化しても、データを保護し、同期し、検証することは難しかったです。

何故なら、コンピューター上の記録は、あまりにも簡単に改変することが出来るからです。そして、データの権限、出典は、独立した数えきれない程のシステムへ分裂し、またがって存在しています。

ブロックチェーンは、データを承認可能で、個別に監査可能とする分散形のメカニズムを提供しました。
ビットコインのブロックチェーンは、最も信頼されている改変不可能なデータストアです。しかしながら、ビットコインのトランザクションに関係が無いデータを扱うことについては、有用とは言いがたいものです。
Fこの文書では、我々はどのようにしてFactomが分散形で自律的なプロトコルを設計し、ビットコインのブロックチェーンを暗号通貨としてのビットコインから効率的に切り離し、actomは、事業者や開発者が、通貨としてのプロトコルに溺れてしまうことなくブロックチェーンのテクノロジーにアクセスできるようにする手段を提供します。

分散形の自律的なシステムを制作したかを説明します。

クライアントによって定義される鎖であるエントリー

エントリーのクライアントサイドでのバリデーション、
エントリーを記録するための分散形の承認アルゴリズム、
そしてセキュリティ向上のための、ブロックチェーン・アンカーリングの方法について議論します。

仕様のゴール

Factomは、より速く、より安く、ブロックチェーンの肥大化をもたらさない、ブロックチェーンを基盤とするアプリケーションの開発方法をもたらします。サトシ・ナカモトがビットコインのブロックチェーンを始めたとき、彼はトランザクションが記録される方法に革命をもたらしました。これまで決して永続的で分散されていた、信頼の必要ない、記録の台帳は存在しませんでした。
そして、開発者は、この記録の台帳の上にこぞってアプリケーションを作ろうとしました。

しかし、不幸なことにビットコインは現在いくつかの核となる
本質的な下記の制限にぶつかっています。

1)スピード

分散型で、ビットコインで使われているPoWの合意に基づく設計であるがために、10分の承認時間を保ち続けるように、難しさが設定されています。

高いセキュリティを望むアプリケーションでは、複数の承認が必要とされるかもしれないです。一般的に必要とされる承認は、6回の承認であり、1時間以上も待たなければなりません。

2)コスト

デフォルトのトランザクション費用は、およそ0.01mBTC(2014年11月現在凡そ$ 0.003USD)です。
BTCの交換レートはこれ迄ずっとボラティリティが高いものでした。もしBTCの値段が上がれば、トランザクションの費用も上がるかもしれません。このことは、非常に多くのトランザクションを解決しなければならないアプリケーションにとっての深刻なコスト面での障害となってしまうかもしれません。
加えて、ブロックサイズの制限と、報酬の分配に対する制限といった数多くの事実は、トランザクション費用の増大をもたらすかもしれません。

3)肥大化

ビットコインのブロックチェーンのサイズはブロックごとに1MB制限されています。トランザクションのスループットは、1秒当り7トランザクションに制限されています。
ブロックチェーンを使って情報を書き込み保存しようとするあらゆるアプリケーションは、このトラフィックの中に加えられるでしょう。
様々な関係者がブロックサイズの制限を増大させようとするがために、この問題によって、人為的、政治的に費用が加算されるかもしれません。

Factomはこれらの3つの核となる制限に取り組みます。
Factomは、通貨のトランザクションを超えた、関数と機能を提供するアプリケーションのためのプロトコルを作ります。
Factomは、アプリケーションが、速く、安く、ビットコインを肥大化させる

こと無しに動くようにするための、標準的で効率的なセキュアな基盤を築きます。

Factomのエコシステム
いくつかの第一義的なFactomのエコシステムの構成要素を下記に示します。一旦システムが出来上がれば、Factoidsの発行やユーザーアカウントを含めて、Factom，ビットコイン、ユーザー間とで下記のような作用を主としてトークンの価値は転送されます。

1. Application OwnerがFactoidと共にエントリークレジットを購入する。

2. アプリケーションはエントリーを記録する
3．Factomのサーバーは、エントリーブロックと、ディレクトリブロックを生成する。
4. Factomは、ブロックチェーンへのディレクトリブロックのアンカーを保護する

下記のセクションで、これらのことの詳細を記述します。

セキュリティと証明

どのようにしてFactomはエントリーを保護するか

Factomはビットコインの貨幣の交換の外側でイベントを記録する機能のセットを拡張します。Factomは、永続的なEntriesに加えることための最小限のルールセットを持っています。

Factomは、大半のデータのバリデーションのタスクをクライアントサイドで行うようにしています。

唯一Factomが実行するバリデーションは、Factoidを交換すること、Factoidをエントリークレジットへと変換すること、そしてエントリーに対して適切に支払われ、記録されたことを確認することです。

Factomは、ネットワークを走らせるトークンのインセンティブに関して僅かなルールをもっているが、そのチェーンを使っているユーザーによって記録された文書の正当性を確かめることは出来ません。

ビットコインはトランザクションをいくつかのインプットから、いくつかのアウトプットへ価値を転送する機能に制限しています。一般的にある署名を求めている、インプットビットコインのスクリプトはシステムがその正当性を保証するために十分です。

このことは、検証のプロセスを自動化し、監査のプロセスを簡単にすることとなるでしょう。もしFactomが使われれば、例えば実際の不動産の転送を記録することに使え、Factomはそのプロセスが起こったことをシンプルに記録するでしょう。不動産が移る手続きはとても複雑です。例えば、地方の権力が、不動産についての特別な要求を持っているかもしれません。もしバイヤーが外国人ならば、もし農民ならば、もし一時的居住者ならば、などといった条件に対してです。不動産の権利は、位置情報や、価値や、機構といったことに基づく数多くのカテゴリに分類されるかもしれません。それぞれのカテゴリは、スマートコントラクトに対してのバリデーションのプロセ

スを反映して、独自のルールを持つことが出来ます。この不動産の例の場合、完全に所有権の意向正当性を確認するには暗号的署名単体では不十分です。Factomはそのような場合、移管を検証するよりもむしろ、起こったプロセスを記録することに使われます。

ビットコインのマイナーは、2つの主なタスクを果たす。先ず第一に、ダブルスペンド問題（2重支払い問題）を解決します。それは2つの衝突するトランザクションが同じ資産を二度使っていないかを確認することです。そして、どちらが許容されるべきかを解決します。

2つ目のマイナーが果たす仕事は、(フルノードとともに)監査することです
ビットコインのマイナーは、正当なトランザクションのみを含め、既に監査されたブロックであると考えられるブロックのみをブロックチェーン内に含めます。薄いクライアントでは、ビットコインの完全な記録を必要としていないものの、受け取った価値が既に使われているかどうかを確かめます。(SPVを参照)

どのようにしてFactomのサーバーと監査のサーバーがエントリーをバリデーションするか?

Factomはビットコインのマイナーが行う2つの役割を2つのタスクへと分けます。
1. エントリーを最終的な並び方で記録すること

2. 正当性の保証のためにエントリーの監査を行うこと

1 – Factomのサーバーはエントリーを受入、ブロックのような状態へとまとめます。10分ごとに、エントリーの順番は、ビットコインのブロックチェーンへアンカーを挿入することに依って、不可逆的な状態へとされます。
Factomは、データのハッシュを10分に渡って生成し、そのハッシュをブロックチェーンに記録することに依って、このことを行います。

2 – エントリーの監査のプロセスは、トラストを用いる、もしくは用いないどちらの方法でも行えます。監査は非常に重要であり、何故ならFactomは、Factomのデータ・セットに入る前には、エントリーをバリデートすることが出来ないためです。

信頼に基づく監査では、軽量クライアントは、彼らが選んだ監査の能力を持った監査役を信頼します。監査役は、エントリーが正当であることを、エントリーがシステムに挿入された後に確かめます。監査役は、自らが暗号的な署名をしたエントリーを提出します。その署名は、監査役が行うべきであると求められた全てのチェックを通過した証明です。そして上記のような監査の必要条件は、Factomのチェーンの一部となることが出来ます。
先ほどの不動産の例の場合には、監査役はその移転がローカルの基準にそっているかどうかをダブルチェックします。
監査役は、公にその移管が正当であると証明します。

信頼のいらない監査のプロセスは、ビットコインと似ています。
もしシステムが、ビットコインの正当性の数学的な定義と内部的に一貫しているならば、プログラム的に監査されます。

もし転送のためのルールがコンピューターに依って監査可能であるならば、アプリケーションは、関連するデータをダウンロードし、自らで監査を行う事が出来ます。そのアプリケーションは、システムの状態に注意した状態でビルドすることができ、どのエントリーが正当かそうでないかを決めることが出来ます。

Mastercoinや、Counterpartyや、Colored Coinsは、同様の信頼の仕組みを持っています。これらの全てのクライアントサイドでのバリデーションのプロトコルは、トランザクションがビットコインのブロックチェーンに埋め込まれているかどうかを示しています。ビットコインのマイナーは、正当性のために監査することは出来ません。そのため、プロトコルに沿っているかのように見える不正なトランザクションが、ブロックチェーンに挿入されることが出来ます。

クライアントは、ブロックチェーンを通じたこれらのプロトコルによるスキャンの一つをサポートし、潜在的なトランザクションを見つけ、正当性と、アセットのコントロールがある場所の解釈を構築します。(通常はビットコインアドレスのある場所です)。これらのプロトコルの下で、独自の監査のプロセスを設計するかどうかはクライアントに委ねられています。

Factomではこれらのクライアントサイドで承認されたプロトコルを動かすことは、プロトコルごとにトランザクションを定義し、トランザクションを持っているチェーンを生成することを意味します。そのトランザクションのプロトコルは、Factomのものというよりもむしろビットコインのプロトコルです。しかし、Factomでは、ある特別な方法を用いてビットコインのトランザクションの中にエンコードすること無く、必要な情報を簡単に表現することを可能としています。

否定の証明

ビットコイン、土地の登記、その他多くのシステムにおいて、根本的な問題を解決しなければなりません。それは否定の証明（Proving a negative)です。これらは、何かの”物事”がある人へ移管したことを証明し、他の誰かには移管されなかったことを証明することです。制限のないシステムの下では否定を証明することは不可能であるが、制限のあるシステムの下では可能です。暗号通貨は、この問題をトランザクションが見つかったときに限ることに依って解決しました。
ビットコインのトランザクションは、Bitcoinのブロックチェーンの中でのみ見つかります。もし関連するトランザクションが見つからなければ、それはビットコインのプロトコルの観点から考えると、存在していないと考えられ、そのBTCは2度送られていないと考えられます。（2重支払い）

所有されている特定の土地の所有権はよく似ています。政府のレジストリで土地の移管を記録するシステムと、法的なシステムを考えると、記録されていない移管は不当であると考えられています。(訴訟が無かった場合)
もし個人が、その土地が誰も持っていない状態であるかどうかを確かめたいのであれば（例えば他の誰もその土地の所有権を主張していない時）、その回答を得るために政府の記録見に行くことになるでしょう。そして、その個

人は、政府の記録を使って否定の証明が出来ます。つまり、その土地が第三者によって所有されていなかったということです。

所有権の登録が行われる場所においては、必要とされていない。政府のレジストリにおいては、何が登録されているかだけが主張されている。しかし、その政府のレジストリの登録が正当でなくなるような個人的な移管はとても良く行われているかもしれません。

上記のどちらの場合においても、あるコンテクストの中では否定を証明することが出来ます。マスターコインの場合にはとても強いものです。土地のレジストリでは、そのレジストリのコンテクストに限って、開けた試練かもしれません。現実の世界は煩雑であるため、Factomは、デジタルアセットの正確さだけでなく、現実の世界の時折煩雑な状態を収容できるように作られています。

Factomでは、データの分類による階層化があります。Factomは、チェーンのエントリーのみを記録します。様々なユーザーによって定められたチェーンは、Factomがプロトコルのレベルで行っているように依存関係を持っていません。このことは、ビットコインではあらゆるトランザクションが潜在的にダブルスペンドの危険があるため、バリデーションされなければならないこととは違う点です。Factomは、チェーン化されたエントリーを組織することに依って、全てのFactomのデータが1つの台帳に書き込まれることなく、アプリケーションがより小さな検索範囲で済むようにしています。

もしFactomが土地の移管を管理するために使われるのであれば、あるアプリケーションは、そのようなレジストリを記録するためにチェーンを使い、他のチェーン、例えばセキュリティカメラのログを記録するために使われているチェーンを依存関係なく安全に無視するでしょう。

政府の裁判が土地の登記を変更するために行われたとしたら、関連するチェーンはその結果を反映するために更新されるでしょう。そして、その歴史は決して失われません。そのような変更が、実際に法的、もしくは他の観点からみて実際に不当であるとしても、Factomの中でイベントの順序を隠すために記録を変更することは出来ません。

Nick Szaboは資産のクラブについて文書を書いています、それはこのシステムと多数の重なりが有ります。ここに彼の”Secure Property Titles with Owner Authority”からの抜粋を書きます。

“While thugs can still take physical property by force, the continued existence of correct ownership records will remain a thorn in the side of usurping claimants.”

“悪者が物理的な資産を力づくで奪うことが出来ても、
正しい所有権の記録の継続的な証明が有ることで、奪ったという宣言の側への刺が残る”

どのようにしてアプリケーションはFactomのチェーンの正当性を証明するか

Factomでは、エントリーの承認は行いません。代わりにエントリーはクライアントサイドでユーザーとアプリケーションによって承認されます。アプリケーションは、チェーンが従うべきルールが分かる限りは、不当なエントリーの存在が、解釈できなくなる崩壊を引き起こすことはありません。チェーンの中のエントリーは、アプリケーションによって、考慮されていないルールには従いません。

ユーザーは、チェーンのためのルールのセットを用いて、チェーンのユーザーに対してのルールとあらゆるしきたりで更新することが出来ます。最初のチェーンへのエントリーはルールのセットと、監査プログラムのハッシュを保持します。例えば、これらのルールのセットはFactomに対応して動くアプリケーションが、不当なクライアントサイドのエントリーを無視するために解釈されるかもしれません。

強制された結果について、特定することが出来ます。特定の強制された結果の必要条件を満たさないエントリーは拒絶されるでしょう。しかしながら、そのルールや、監査のプログラムによって拒絶されたエントリーは、未だ記録されるかもしれません。この種チェーンのユーザーは、このタイプのチェーンの結果を承認するために監査のプログラムを走らせる必要があります。Factomのサーバーは、監査プログラムを使うルールに対してのバリデーションは行いません。

アプリケーションの正当性は（ユーザーに依って定義された鎖と共に）、Factom上に書かれたアプリケーションに対して多数のアドバンテージをもたらします。

1. Factom上のアプリケーションは、アプリケーションに取って意味のある情報どのようなエントリーも挿入することが出来ます。そのため、アカウントのリストを認証するハッシュのリストは、アセットの交換と同様に簡単に記録することが出来る。

2. ルールの実行はとても効率的です。分散形のネットワークは必ず正当性の確認のルールを実行しなければなりません。そして正当性は全てのノードが全てのバリデーションを行うことを必要としています。クライアントサイドのバリデーションは、クライアントアプリケーションを走らせるためのルールにのみ気にするシステムだけが必要です。

   Factomでは、どのような言語や、プラットフォームや、外部のデータを仕様作成者が選んでいようとも、チェーンがルールを決定し出来るようにしています。アプリケーションの一部においてのこれらの決定は、どれも他のアプリケーションに影響を与えるものではない。

3. Factomのサーバーは、記録されたエントリーに対しての知識をほとんど持っていません。我々は知っている内容を制限するために commitment scheme を用いている　そこでは、エントリを記録するためのコミットは、どのようなエントリーかということに先立って生成される。このこと

によって、エントリーを記録するためのFactomの役割はとてもシンプルで、個別のサーバーのプロセスを公のものとします。

Factomのサーバーはネットワークのフルノードからの情報を受け入れている。そして決定と行動はいつでも見ることが出来ます。プロセスが行われていないことも、Factomの外部、そして内部から見るネットワークから監査することが出来る。個別にFactomのサーバーがエントリーの記録の責任を果たしているかどうかを確認することは簡単で、Factomは潜在的にエラーとなる行動を隠すことは出来ます。

4. 速度を記録することはとても早く、何故ならばFactomのサーバーによるチェックの数は最小となるようになっているためです。

5. Factomにおいての特定のチェーンに対しての証明は、他のチェーンの知識を必要としない。ユーザーは、ユーザーが使っているFactomのセクションだけが必要であり、他のセクションは無視することが出来る。

どのようにしてFederated Serverはチェーンを管理するか

Factomは分散形の集約、パッケージ化、そしてブロックチェーンへデータを安全に保管する方法を持っている。FactomはこのことをFederated serversのネットワークに依って成し遂げている。これらのサーバーはシステムの異なる面に対しての責任を

どの一つのサーバーも、全体のシステムを支配せず、ただシステムの一部である。そしてどのサーバーも永久的にシステムの一部であることはない。それはFactomのサーバーが毎分Factomのサイクルの一部となって交代することである。
Ferderated サーバーは、ユーザーのチェーンの一部に対して責任を持って、ディレクトリーブロックが作られて始まったときからプロセスは下記のように働きます

1. 全てのサーバーは、プロセスのリストを空にする。

2. ユーザーは、エントリークレジットと紐づく公開鍵を使ってエントリーペイメントを提出する。

3. エントリーに対して支払われるために使われた公開鍵に基づいて、サーバーのうちの１つが支払いを受け付ける。

4. そのサーバーは支払いを受け付けたことをブロードキャストする。

5. ユーザーは、支払いの受入を確認し、エントリーを提出する。

6. エントリーのチェーンIDに基づいて、サーバーのうちの1つはエントリーをプロセスのリストに加える。そしてチェーンIDに対応する適切なエントリーブロックへエントリーを加える。（もしエントリーブロックに対しての最初のエントリーであるならば、生成する。）

7. サーバーは、エントリーのインデックスが並んだプロセスリストと、エントリーのハッシュ（これは支払いに紐付いている）と、サーバーのプロセ

スリスト迄の連続的なハッシュとを含んだエントリーの承認をブロードキャストする。

8. 全ての他のサーバーはサーバーのプロセスリストをアップデートし、リストを承認し、チェーンIDに対応するエントリーブロックを更新する。

9. ユーザーが関係するプロセスのリストの正当性を確かめる事が出来るまでは、関係するプロセスのリストはエントリーを保持する。それにより、Factomの中にエントリーを挿入することが成功したことに対して確信を持つことが出来る

10. 最後に、全てのサーバーはプロセスリストの高さを確かめ、決定論的な秘密の番号（Reverse Hash、即ち長いハッシュのチェーンの連続するイメージ）と、プロセスブロックの連続したハッシュを公開する。（それはプロセスリストの最後のアイテムとマッチする）

11. 全てのサーバーによって定義された、全てのエントリーブロックから、ディレクトリブロックは組み立てられる。そして、それぞれのサーバーは全てのエントリーブロックと、ディレクトリーブロックと、全てのエントリーを持つこととなる。

12. Reverse Hashの集合は、次のラウンドのためのサーバーの中でチェーンIDを再配置するためのSeedを作成するために結合される。

13. 10番目のディレクトリブロックが終わると共に下記を実行する。

    a. エントリーブロックの最後の瞬間にマークルツリーがチェーンIDによってソートされた形で生成される。

    b. 最後のディレクトリブロックを生成し、そのマークルツリーを生成する。

    c. 10個のディレクトリーブロックのマークルルートのマークルルートからアンカーを生成する

    d. サーバーのReverse hash は、アンカーをビットコインに対して書き込むためのサーバーを選択するシードを生成するために組み合わされる。

14. 繰り返す（1に戻る）

Federated サーバーは、自らが責任を持っているチェーンに対してのプロセスのリストを分毎に構築します。

ディレクトリーブロックを最後の時に作るために使われるエントリーブロックを構築するのと同様に構築します。
プロセスのリストは、残りのネットワークに対してのサーバーによって行われるブロードキャストの決定にとって重要です。

Federated サーバーは、毎4時間ごとに格付けされます。そのランキングはユーザーによる投票のシステムで決まります。投票のためには、ユーザーはFactom上にプロフィールのチェーンを作らなければなりません。そのプロフィールは、署名された公開アドレスのエントリーの多くの数を含んでいます。ユーザーの投票の重み付けは、プロフィールの公開鍵によって定められます。

プロフィールの公開鍵の重み付けの関数のように為されます。

・最後の6ヶ月間に購入されたエントリークレジットの重さを量ること。

・最後の6ヶ月間に作られたエントリーの数を量ること。(各月に作られたエントリー、現在の月を6、一ヶ月前を5として重み付ける)

Nのサーバーによって運営された時には、Nサーバーのトップランキングがfederated サーバーとなります。そして他のnが監査のサーバーとなります。全てのサーバーは投票によるランキングに基いて、維持管理されます。Nの数は最初は16として設定されているが、共同体の議論によって変わりえます。また、Nの数は、トランザクションのボリュームによっては小数の値もありえます。全てのサーバーは、毎回のハートビートの期間ごとに、ハートビートをブロードキャストしなければなりません。（エントリーの証明のブロードキャストは、ハートビートとなります。）

もしサーバーがハートビートを受け取らなかったか、エントリーを承認することがタイムアウトの時間以内に出来なかった場合には、サーバーはSFM(Server Fault Message)をブロードキャストします。もし多数がSFMをブロードキャストした場合には、federated サーバーは”Failed”の状態と見なされ、監査のサーバーへと左遷されます、そして次のランキングの監査のサーバーが取って代わります。ビットコインの伝播の時間の感覚をみれば、4時間ごとにハートビートが行われ、そのタイムアウトは8秒ごとです。そしてFactomのコンセンサスについて更に詳細をいえば、そのアルゴリズムである”Factom Concensus”の文書を参照して下さい。

Factomシステムの概要

Factomはレイヤー化されたデータのセットから成り立っています。

Factomは、最も高いディレクトリーブロックによって、階層化された一連のブロックから成り立っている。これらはミクロなチェーンを構成し、コンパクトな参照を持っている。サイズを小さくするためにディレクトリーブロック内にあるそれぞれの参照は、エントリーブロックとチェーンIDのただのハッシュです。これらのエントリーブロックは、特定のチェーンIDを持っています。Factomシステムにおけるレイヤーとコンセプトは下記です。

1. ディレクトリレイヤー:　エントリーブロックのマークルツリーを管理します。

2. エントリーブロックのレイヤー:　エントリーへの参照を管理します。

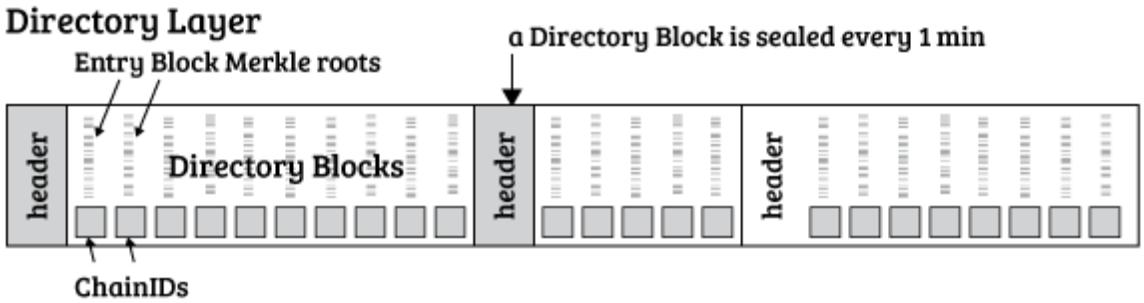
3. エントリー:　アプリケーションの生のデータを含み、プライベートなデータのハッシュを持っています。

4. チェーン: 特定のアプリケーションに対するエントリーの組です。

ディレクトリレイヤー: どのようにしてディレクトリレイヤーはマークルツリーを組織するか

ディレクトリレイヤーは、Factomシステムの第一位の階層にあり。ここではディレクトリブロックに依ってカバーされます。

時間の間中に更新されてきたエントリーチェーンのIDを定義しています。(チェーンIDはユーザーのチェーンのエントリーを同定し、そのチェーンIDの生成は後に議論します。そしてチェーンIDとチェーンIDのデータを含むエントリーブロックのマークル・ルートとのペアの組から成り立っています。



それぞれのエントリーブロックは、ディレクトリブロックの中で参照されており、64 bytesの長さを取っています（32bytesずつチェーンIDと、エントリーブロックのマークルツリーに対して）何百万というエントリーも凡そ64MBのサイズのディレクトリーブロックとなります。もし、平均的なエントリーブロックが5つのエントリーを持っていれば64MBのディレクトリーブロックは、高位の500万の特定のエントリーへの管理を提供するでしょう。もしアプリケーションがディレクトリーブロックしか持たない場合、全てのエントリーブロックをダウンロードすることなく関心があるエントリーブロックを見つけることが出来る。個々のアプリケーションはチェーンIDの小さな組だけに関心を持つことが出来ます。

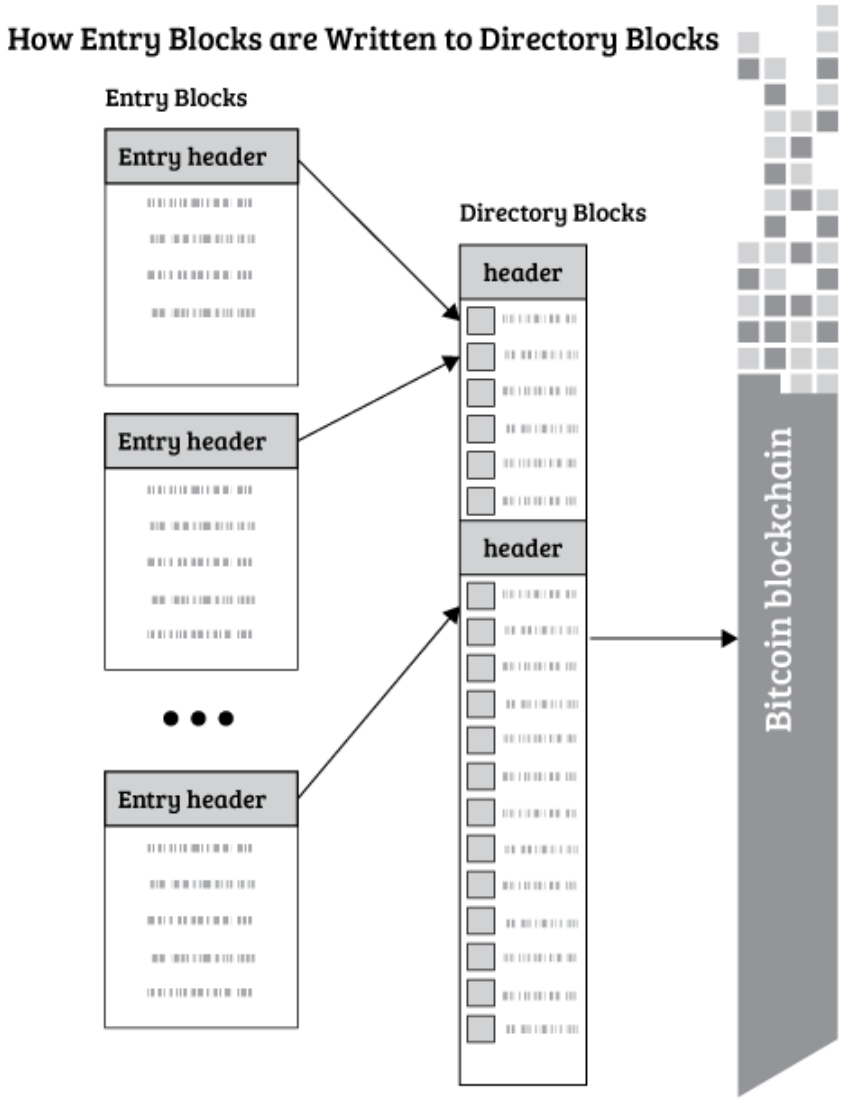
このことは個人のクライアントが記録のシステムとしてFactomのために必要な通信料を大きく制限します。例えば不動産の移管を監査するためのアプリケーションはビデオカメラのセキュリティログを無視することが出来ます。Factomのサーバーはエントリーブロックのマークルツリーを集め、それをパッケージ化してディレクトリーブロックに挿入します。10の連続するディレクトリーブロックがマークルツリーを通してハッシュされると、マークルルートはビットコインのブロックチェーンに記録されます。このことは、blockchainの膨張を最も小さいものとしながら、Bitcoinのハッシュパワーによるセキュリティを維持し続けることが出来ます。マークルルートをビット

コインのブロックチェーンに挿入することを我々はアンカーリングと読んでいます。 “Appendix: Timestamping into Bitcoin”に詳細があります。

ディレクトリーブロックに挿入されたデータは、通信料と、ストレージの観点からみて、最も高価です。
全てのFactomのユーザーは、チェーン内にデータを見つけるためにそのチェーンが始まった時からのディレクトリーブロックのフルセットが必要です。
ディレクトリーブロックのサイズを増やす活動は、個々のチェーンの生成と、最初に個々のチェーンを更新する時です。

これらの活動、よく出来た団体（ビットコイン）に対してコストを外部化しようとする行動です。そのため、これらの活動を行うアプリケーションは、シンプルなエントリーよりも多くのエントリークレジットを消費しなければならなくなり、ディレクトリーブロックの肥大化が防がれるようになっています。

エントリーブロックレイヤー:
どのようにしてエントリーブロックレイヤーがハッシュとデータを組織して扱うのか



エントリーブロックは、Factomのシステムで二番目の階層に位置する層です。エントリーブロックにおいて、個人のアプリケーションは、チェーンIDから他の関連しうる全てのエントリーを見つけて、検索範囲を拡張することが出来ます。

1つのエントリーブロックは更新されたチェーンIDをディレクトリーブロックごとに持っている。そのエントリーブロックは、個々のエントリーのハッシュを含んでいます。

エントリーのハッシュ達は、データの存在を証明し、そして分散したハッシュテーブルのネットワークにおいて(DHT)、エントリーを見つける鍵を与えてくれます。（詳細は”The factom Peer to Peer network” を参照）

そのエントリーブロックは、チェーンIDと関連する可能性あるエントリーの最大の規模を内包しています。もしエントリーがエントリーブロックの中で言及されていなかったならば、それは存在しなかったものとしてみなされます。このことによって、先ほどセキュリティと証明のセクションで説明した

ように、アプリケーションにとって、エントリーが存在しないことを証明することが出来るようにしています。

どのようにしてディレクトリーブロック上にエントリーブロックが記載されるか

エントリーブロックは、ChainIDに関連する可能性のある全てのエントリーを内包しています。もしエントリーがエントリーブロックに参照されていなければ、そのエントリーは存在していないと考えられます。これにより、”セキュリティと証明”のセクションで述べたように否定を証明する事ができます。

エントリーブロックでは、意図的にエントリーそのものは含んでいません。このことによって、全てのデータが一緒にまとめられているよりも、エントリーブロックはかなり小さくなります。
また、エントリーをエントリーブロックから切り離すことによって、監査者が監査を行う事を簡単にしています。

監査者は、共通のチェーンにおいて、エントリーを承認ないし拒絶する個別のチェーンにエントリーをポストすることが出来ます。そして、その監査者はエントリーにおいて、拒絶の理由を付け加える事が出来ます。もしアプリケーションが監査者を信じているならば、監査者がエントリーを承認したか拒絶したかの相互の参照を持つことが、エントリーの中身を知ること無しに出来ます。

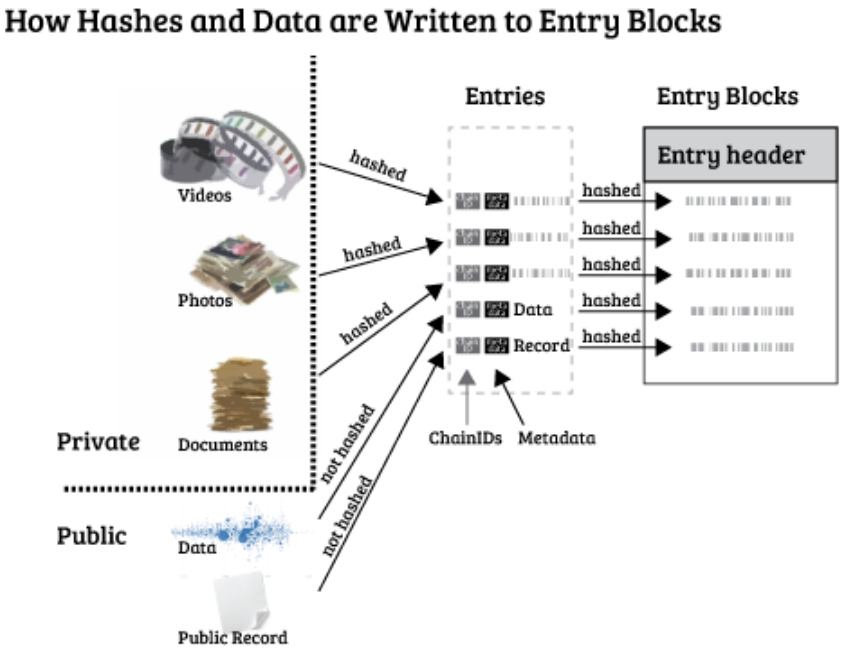
そしてアプリケーションが、監査を通り、多様な監査者が同じエントリーを参照出来ます。

そしてそのエントリーは分散形のハッシュテーブルにおいて、ただ一度だけ存在しています。

エントリーは、ハッシュが占める32バイトよりもかなり大きくなることを期待されています。
無視されることは、アプリケーションが、必要でないと知っていて参照されない全体のオブジェクトを保つ必要はありません。

あるエントリーは、土地の移転の特徴の詳細を述べます、そして、土地の移管は、見つかるように期待されています。1人や、より多くの監査者は、自らのチェーン内にある土地の移管に対応するハッシュを参照することが出来、暗号的な署名を通過したか失敗したかを示すために追加することが出来る。その土地の移管についての文書は一旦保存されれば、他の数多くのチェーンからも参照されることとなります。

エントリー： どのようにしてエントリーは生成されるか



エントリーは、ユーザーに依って生成され、ファクトムに送られます。情報をハッシュするかエンコードすることに依って、ユーザーはエントリーのプライバシーを保証できます。エントリーは平文でも挿入することが出来、必ずしもエンコーディングや、匿名化は必要では有りません。ドキュメントのハッシュを記録することに依って、Factomは基本的な公開記録の証拠を提供します。

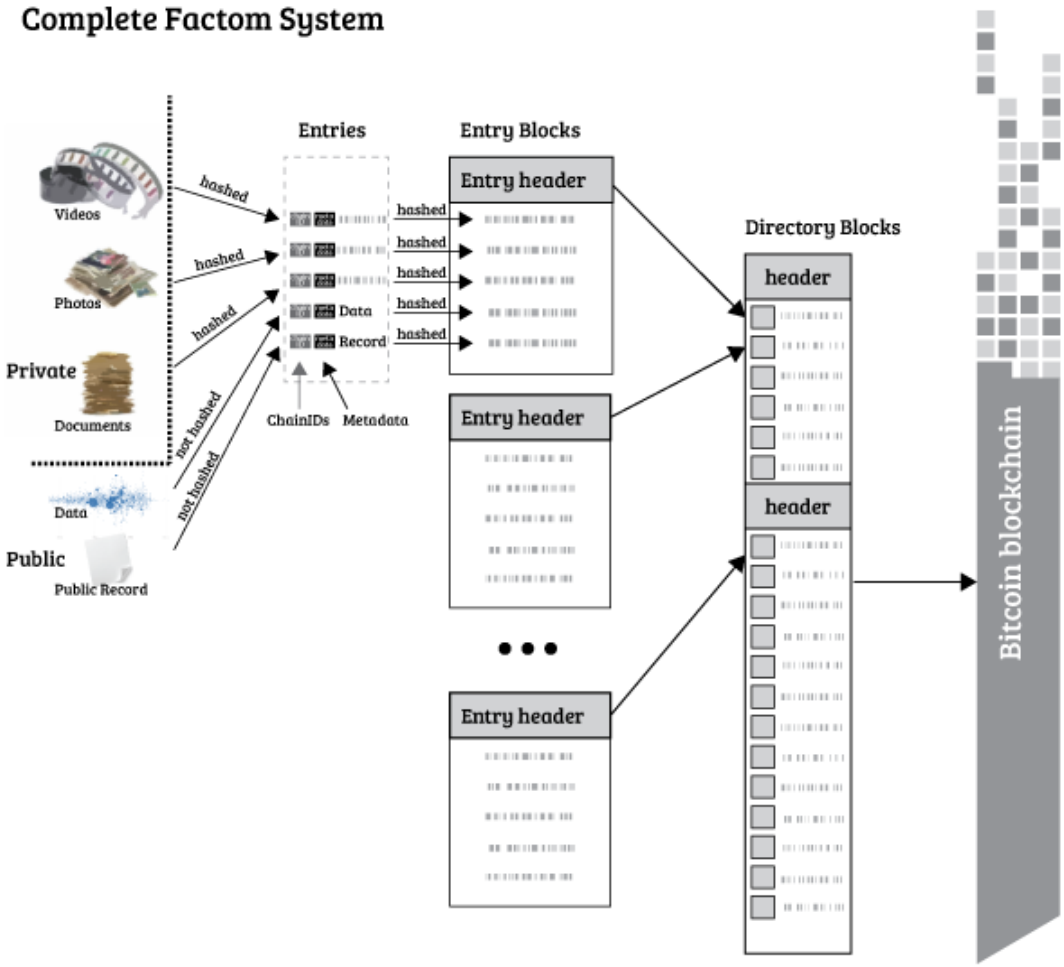
ハッシュを作成し、過去に記録したハッシュと比べることが、その文書を提出することによって、出来るようになります。受け入れられるデータには数多くの柔軟性があります。それはハイパーリンクのようなものです。あまり大きくはありませんが、それでも比較的大きいものです。これはビットコインとよく似ています。100kB以上のビットコインのトランザクションは可能ですが、比例する大きなトランザクションのフィーを払わなければなりません。このサイズはビットコインではかなり大きいですが、Factomでは控えめなサイズです。あらゆるビットコインのフルノードが全てのブロックチェーンが正当であることを証明しなければならないため、サイズは小さい必要があります。しかしFactomでは、あるチェーンを全てバリデートするために必要なのは、最も高位のディレクトリブロックだけです。もし特に興味があるわけで無いチェーンのデータがあっても、人はそのチェーンをダウンロードしないでしょう。

簡単な編集不可能なTwitterのようなシステムの例を上げます。あるセレブは、ある文章の一片としてエントリーを作り。そして秘密鍵で署名することに依って、セレブ自身が呟いたものであることを示します。

そのセレブのフォロワーは、どのチェーン上にパブリッシュされたかを知ることが出来、更新情報をモニターすることが出来ます。あらゆる新しい署名されたエントリーは、フォロワーのアプリケーションによってTweetとして

認識されます。他の人のTweetが、そのセレブのチェーンにエントリーとして追加されることによって、他の人がそのセレブに対してTweetすることも出来ます。

チェーン: どのようにしてエントリーはチェーンの中に組織されるか



Factomにおけるチェーンは、アプリケーションに関連するエントリーの一連です。これらの一連のエントリーは、Bitcoin2.0のコアとなる情報です。チェーンは、これらのイベントの結果を文書化し、イベントが起きた結果についての監査証跡を提供します。更に暗号学的な署名を加える事に依って、それらのイベントはよく知られているソースに基づいた証拠となるでしょう。

チェーンはディレクトリブロックとエントリーブロックの中で配置されたデータの論理的な解釈です。ディレクトリーブロックは、どのチェーンがアップデートされたか、そしてエントリーブロックはどのエントリーが追加されたかを示します。どのようにビットコインのフルクライアントが、ローカルなUTXOを維持しているのかの類推として考えることができます。

そのUTXOは、現在のところブロックチェーン内にあるわけではなく、フルクライアント自体が解釈しています。

FactomのP2Pネットワーク

Factomは2つの目的のためにP2Pのネットワークを持っています。コミュニケーションとデータ保管のためです。

Factom P2Pのコミュニケーション

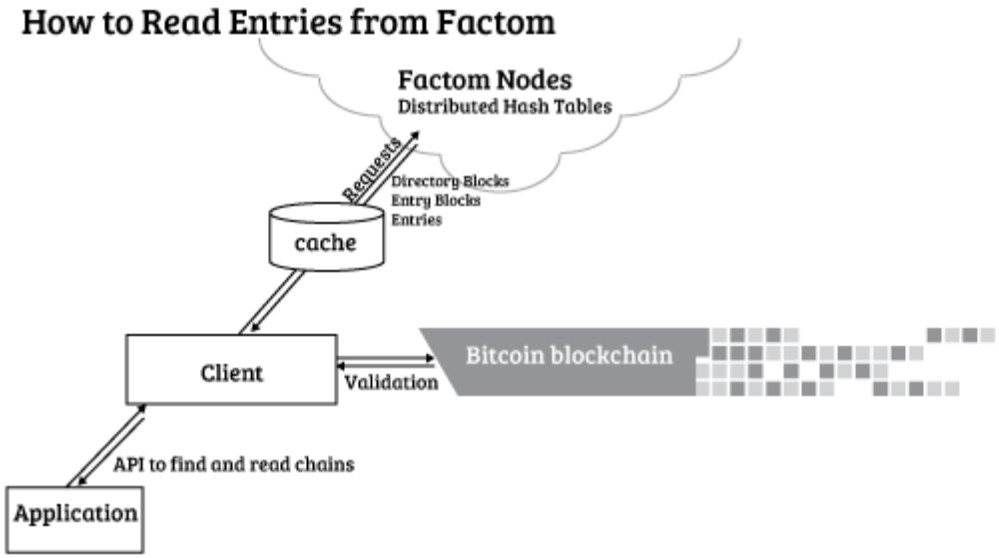
Factomはビットコインと非常によく似ているP2Pのネットワークを持っています。全てのFactomのデータを持っているfullノードから成り立っています。フルノードは、ネットワークを通して正しいデータを埋めるための網目状のネットワークを生成します。そのFederated サーバーは、フルノードですが、全てのフルノードがfederated サーバーであったり、監査のサーバーではありません。このことはビットコインと非常によく似ています。ビットコインでは、マイナーはフルノードですが、全てのフルノードがマイナーでは有りません。このことによって、Federated serverの個々に対するDDOSの能力を制限します。彼らはネットワークの内側のどこでも繋がる事が出来、データの構造を作るために必要なデータを手に入れることが出来ます。サーバーがコンセンサスに達し、署名せれたデータに対して、P2Pネットワークを通じてデータを公開します。そのP2Pの波は、FederatedサーバーがIPアドレスに基づいた検査出来る能力を制限します。何故なら、繋がろうとしているノードによって、トラフィックはミクシングされるからです。このことは監査を制限し、監査のサーバーがエントリーを見ることが出来、そのことによって、エントリーブロックに含まれるエントリーを監査のサーバーが見ることが出来るようになります。

監査のサーバーはFederated サーバーとなるための投票があるため、監査のサーバーは、悪意のあるコードを公に公開するインセンティブをもっています。

データの保存と散布

Factomのデータ構造は(ディレクトリーブロック、エントリーブロック、エントリー)で、Factomが有用であるために必要です。これらはパブリックであり、2つの場所に保存されます。Federated サーバーと、監査のサーバーで、正しい判断を新しいエントリーの追加に対して行うために、データを維持する必要があります。それらのサーバーは上記のデータを保持しており、フルノードのサービスとして、サービスとして提供します。ノードの一部でもあり、特定のアプリケーションに関係するデータのみを共有する特定のノードです。ノードの一部のためのPeerの発見は、分散形のハッシュテーブルによって行われます。

どのようにしてエントリーをFactomから読み込むのか

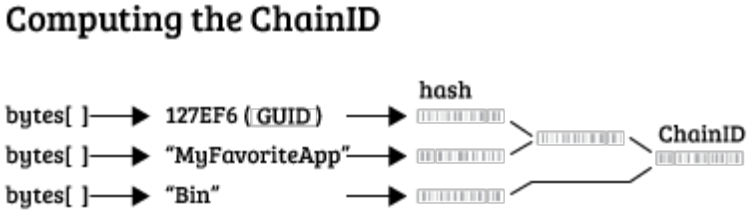


分散形ハッシュテーブルのセットアップに依って、Factomのデータ・セットの全体は完全なデータ・セットが掌握できないサイズに大きくなってしまった場合でさえも、効率的なピアーへのデータの分散を可能としています。その分散形のハッシュテーブルは、データが独立してあるFederated サーバーもしくはフルノードで独立して保存される事を可能としている。喩え、全てのフルノードがインターネットから無くなったとしても、数多くのデータの当事者によって、特定の一部のセットのデータが保持されていることとなります。

より深いFactomについての議論

どのようにしてFactomのチェーンに名前を付けるか

Factomは全てのエントリーをチェーンIDのもとに、グループ化します。そのチェーンIDは、チェーンの名前から計算されて導かれます。チェーンIDはチェーンの名前のハッシュです。



チェーンの名前は、任意の長さのbyte arrayです。下記の表を御覧下さい。チェーンの名前から、チェーンのIDに変換するハッシュのオペレーションと

なっています。それは単純なプロセスです。チェーンの名前をチェーンIDから導くこと簡単ではなく、ルックアップテーブルが必要となるでしょう。

ユーザーはチェーンの名前を与えなければなりません、チェーンIDがハッシュ等の結果として導かれるようになる。このことはハッシュされていないデータがチェーンのIDとなってしまうことを防ぎます。そしてディレクトリーブロックに保存されます。この変換はブロックの構造に平文が挿入されることを防ぎます。

チェーンの名前は全く持って任意です。ランダムの数字や、テキストのストラング、また公開鍵でも構いません。個々のアプリケーションは、異なるチェーンの名前から由来する意味を知ることも出来ます。

1つのありうる変換は、人間が読めるテキストをチェーンの名前とすることで、このことによって、チェーンの構造がロジカルな階層となることを助けます。喩えチェーンが元々は階層的ではなかったとしても。ユーザーは同じ名前を使うことが出来ます。しかし、偶然チェーン同士で交差が起こってしまわないように、シンプルな修正が為されます。

下記のようなパスを考えて下さい。:

MyFavoriteApp/bin

スラッシュがある場所において、他のレベルの階層との変換があります。スラッシュはASCII文字の”MyFavoriteApp”と”bin”とを分けています。これらの2つのストリングはバイトに変換され、そのために様々なオプションがあります。ストリングはUTF-16によってエンコードされることがあり、UTF-16や、UTF-32や、ASCIIや、他にもIBMのEPCDICのようなものによってもエンコードされることが出来ます。これらのエンコーディンクが完全に異なるチェーンIDを同じストリングから導き出します。更に言えば、アプリケーションが、GUIDの数字を、名前の変換の際に最初のバイトアレイとして利用することが出来ます。このことによって、あるアプリケーションのチェーンIDの”場所”が他のものと被ってしまう可能性を、只いくつかのバイトをチェーンの生成の際に使うだけで、無くすことが出来ます。

Factoidを使ってEntry Creditsを購入すること

Factoidsは、主要な内部の希少性のあるトークンであり、システムで動く人への報酬の仕組みとして稼働します。
Factomの中にエントリーを入れる権利は、エントリークレジットを得ることで、獲得出来ます。

Factoidsはビットコインが実装されたのと全く同じやり方で実装されており、例えば、マルチシグや、マルチシグのインプットや、マルチシグのアウトプットが可能です。Factoidのトランザクションは、Factoid用の特別なチェーンによって管理されます。そのFactoidのチェーンは、他のチェーンよりもより制限されて稼働します。Factoidチェーンの中のエントリーはFactoid

のトランザクションとして正当でなければなりません、そうでない場合Factomのサーバーがエントリーを拒絶するでしょう。

Factoidsは、完全に分散形のFactomのプロトコルに含まれており、Factomとビットコインの中での肥大化と、スパムを無くすことが出来ます。Factoidsは、プロトコル内のエントリークレジットに変換する事が出来、Factomのサーバーにはプロトコルから支払われるものです。そのことによってコンセンサスが起こるようになる、何故ならコンセンサスが起きなければ、Factoidsの価値は下落するためです。

Factoidをエントリークレジットに変換するのは、特別な購入のトランザクションをFactoidのチェーンを通じて行うことに依って為される。この購入のトランザクションは下記を含みます。

・ アウトプットが変換されるべきFactoidの総量を支持すること

・ エントリークレジットを受け取るための公開鍵

エントリークレジットは一旦購入されると、他の公開鍵のもとに移管することは出来ません。それらは、エントリーの支払いのためにだけ使用されうか、もしくはFederated サーバーへの投票のために使われる。再販売がエントリークレジットは出来ないため、泥棒にとっての価値を下げます。

エントリークレジットの秘密鍵は、リスクが最小になるように低いセキュリティを持ったエリアに保存される。

エントリークレジットを使って、エントリーを購入すること

Factomのもとにエントリーを追加するためには、希少なリソースを提出することが必要です。そのリソースは、Factoidsから由来するエントリークレジットです。エントリーをFactomに追加するためには2つのステップのプロセスが必要です。
初めにエントリーの追加に対して支払われます。エントリーに対する支払いによって2つのことが達成されます。

1つはユーザーの公開鍵と結びついたエントリークレジットを減少させること。そして、もうひとつはエントリーのハッシュを特定することです。エントリーに対して支払われた後には、サーバーはハッシュされていないエントリーを待って、公開されたら、エントリーを含めます。

1. エントリーへの支払い

- ユーザーによって所有されているエントリークレジットを減少させる。
- 支払いの中でエントリーのハッシュをユーザーが特定する。

2. エントリーの挿入

- ユーザーはエントリーをエントリーブロックへ含めるように公開する。

2段階のステップを踏むことには、様々なメリットがある。1つの利益は、支払いを記録されたデータから分割することです。将来のユーザーが支払いによって生成されたデータをダウンロードしなくて済むようになる。
彼らは、システムを正当化するための最小限のデータをダウンロードすることだけが必要です。それによって、ユーザーは安全に簡単に支払いの情報を無視することが出来る。

他のメリットは、検査に対しての耐久性を得られることです。
エントリーのコンテントを知る前に、Factomサーバーによってエントリーを受け入れることを声明することが出来ます。Adam Backはビットコインについての同様のメカニズムを"blind symmetric commitment for stronger byzantine voting resilience".での投稿において発表しました。もしユーザーか監査するサーバーが適切に支払われたエントリー見せることが出来るのであれば、どのFederatedサーバーも受け入れていないということと、その監査が立証可能であるということです。

エントリークレジットを差し引いているトランザクションは、Factoidのチェーンに似ている特別なチェーンに保存されます。そのFederated サーバーは、有効なエントリークレジットのトランザクションによってのみ、チェーンを満たすことが出来るでしょう

中央のオラクルのサーバーと共に、エントリーのコストを設定すること

Factoidとエントリークレジットの交換レートは、オラクルによって決定されます。コンピュータシステムではオラクルは、システムでは承認されるか、承認されないかを決定できないシステムにとっての情報を提供するプロセスのことです。
オラクルはFactoidsとFactomのエントリークレジットとの交換レートは、ビットコインのトランザクションのコストに比べて1/10から1/100程のコストで済むレートで維持する。

初めにオラクルが中央で実装されるコンバージョンレートがある。Factomは、Factoidのエクスチェンジのレートとその取引総量をFactomのチェーンに記録出来るだけの十分なエクスチェンジを集めた後、交換レートを算出する分散形のコンピューテーションを行う。その定量化のソースのチェーン（取引所に依って維持される）は、様々な取引所の交換レートと、取引のボリュームによってFactoidsとエントリークレジットとの交換レートが決定されます。

そして、購入の後には、エントリークレジットのチェーンは、適切な公開鍵のもとにエントリークレジットを配置し、そのエントリーは下記のようになっています。

・ 公開鍵

・ いくらのエントリークレジットを購入したか

Factoid無しにFactomを使うこと

多くのFactomのユーザーは、ウォレットを必要とせず、あらゆる暗号通貨のアセットを求めないだろう。だが、チェーンを作りたいと思い、彼らのエントリーを追加したいと思うだろう。Factomは、Factoidsの切り分けを考慮するための2段階の記録のプロセスを持っている。

そのプロセスは、クレジットエントリーをFactom上にポストする権利をFactomの交換可能なトークンであるFactoidと切り離しています。

サーバーと他のFactom Tokenの受取り手はエントリークレジットをカスタマーに対してビットコインや、既存のクレジットカードでの支払いなどで販売することが出来ます。
ユーザーはエントリークレジットを持つための公開鍵をサーバーへ提供します。販売者は、適切な量のFactoidsをエントリークレジットに変換し、そのユーザーの公開鍵に紐付けます。
これによって、ユーザーは、Factomサーバーを動かしているFactoidsを持つこと無く、エントリークレジットを購入することが出来ます。規制の観点から見ると、このことは強力です。

Factomのサーバーは、Factoidsをプロトコルから稼ぎます。その当事者は、サーバーとプロトコルだけです。そしてサーバーはエントリークレジットをユーザーに販売します。ユーザーは最終的にFactoidsを残りのシステム全体へと戻す。
エントリークレジットは、移管不可能であり、ユーザーは他のユーザーの公開鍵に対して、一旦割り当てられたエントリークレジットを割り当てることは出来ない、そのため、エントリークレジットの秘密鍵は実用的でも有用でもありません。
そのため、サーバーとユーザーの当事者の間では、どのトランザクションにおいても、トレード可能なFactoidsを取引していません。

最初のエントリーチェーンにおけるエントリー: 自家製チェーンへのサポート

Factomは、支払いと共に最初に権利を主張するユーザーに対して、全てのチェーンにおいて、最初のエントリーを保存します。

最初のエントリーが、そのチェーンの監査のルールを文書化し、その文書、もしくはテキストでのルール、もしくは、チェーンに対しての監査のプログラムのハッシュへのURLを含める場所として通例となることがありえます。この協定は、自家製のものとして解釈される事が出来ます。

秘密のセットを暴きながら、Factomは、チェーンを作り出したユーザーが、最初のエントリーの内容を決定する事を確実なものとする。（The Man in the Middle AttackについてAppendix 2 を参照）

新しいチェーンへのコミットをコールするには、10のエントリークレジットに加えて、1024バイトごとの1クレジットの支払いが必要です。より高い価

格となることで新しいチェーンの生成、即ち外部に掛かるコストが削減される。チェーンを始めるため際に含まれる3つのパラメータが下記だ:

・ チェーンIDのハッシュ

・ （チェーンID+エントリーのハッシュ）のハッシュ

・ エントリーハッシュ

これらの3つのハッシュはエントリークレジットのチェーンに配置される。

一旦チェーンのコミットが受入られると、そのチェーンIDが公開されます。

チェーンの生成が、正当であると公開されるためには、必ずチェーンの名前が与えられなければならない。そのチェーンの名前は、チェーンIDをハッシュに依って生成する。

エントリーはエントリーハッシュを生成しなければなりません。

最後にチェーンIDとエントリーハッシュを結合したものが、コミットの結果と合致しなければなりません。

これらの条件が満たされなければ、Factomのサーバーは、最初のエントリーを記録せず、チェーンは生成されません。

結論

Factomは分散形で自律的なビットコインのブロックチェーン上に有るレイヤーです。Factomのゴールは、殆ど無限に近い範囲のアプリケーションと、その使用例に対してビットコインのブロックのパワーを提供することです。更に言えば、ユーザーが暗号通貨を一切使うこと無く使えるように、Factomは、設計されています。

分散形の改変不可能な台帳は、急進的で、根本的で、予測不可能な、ビットコインのブロックチェーンに代表されるテクノロジーです。その多くの夢は、数学的に正当と示される改変不可能な台帳を受け継ぎ拡張することによって、混沌としている現実世界の交流に対しても使えるようにすることです。

ブロックチェーンに基づく制限のない台帳を構成することを出来るようにすることで、Factomはブロックチェーンの利益を現実世界へと拡張します。

参考文献

“Bitcoin: A Peer-to-Peer Electronic Cash System” Nakamoto, Satoshi. Web. 16 Nov. 2014
https://bitcoin.org/bitcoin.pdf

"Can Blocks Remain Capped to 1MB Forever?" Transactions. Web. 15 Nov. 2014.http://bitcoin.stackexchange.com/questions/18101/can-blocks-remain-capped-to-1mb-forever

"Thin Client Security." - *Bitcoin*. Web. 15 Nov. 2014. https://en.bitcoin.it/wiki/Thin_Client_Security#Simplified_Payment_Verification_.28SPV.29

"Evidence of Absence." Wikipedia. Wikimedia Foundation, 11 July 2014. Web. 15 Nov. 2014. http://en.wikipedia.org/wiki/Evidence_of_absence

"Recording (real Estate)." Wikipedia. Wikimedia Foundation, 14 Nov. 2014. Web. 15 Nov. 2014. http://en.wikipedia.org/wiki/Recording_(real_estate)

"Secure Property Titles with Owner Authority." Secure Property Titles with Owner Authority. Web. 15 Nov. 2014. http://szabo.best.vwh.net/securetitle.html

"Patent US4309569 - Method of Providing Digital Signatures." Google Books. Web. 15 Nov. 2014. http://www.google.com/patents/US4309569

"Block Timestamp." - Bitcoin. Web. 15 Nov. 2014. https://en.bitcoin.it/wiki/Block_timestamp "OP_RETURN and the Future of Bitcoin." - Bitzuma. Web. 15 Nov. 2014.

http://bitzuma.com/posts/op-return-and-the-future-of-bitcoin/ "Goblin/chronobit ." GitHub. Web. 15 Nov. 2014. https://github.com/goblin/chronobit

"How Can One Embed Custom Data in Block Headers?" Mining. Web. 15 Nov. 2014.

http://bitcoin.stackexchange.com/questions/18/how-can-one-embed-custom-data-in-block- headers

"Headers-First Synchronization Coming Soon to Bitcoin Core - CryptoCoinsNews." CryptoCoinsNews. Web. 15 Nov. 2014. https://www.cryptocoinsnews.com/headers-first-synchronization-coming-soon-bitcoin-core/

“Enabling Blockchain Innovations with Pegged Sidechains - Block Stream ” Web. 15 Nov. 2014.

http://www.blockstream.com/sidechains.pdf

"[Bitcoin-development] 2-way pegging (Re: is there a way to do bitcoin-staging?)" / Mailing Lists. Web. 27 May. 2014. http://sourceforge.net/p/bitcoin/mailman/message/32108143/.

"Could the Bitcoin Network Be Used as an Ultrasecure Notary Service?" Computerworld. Accessed 27 May. 2014.

http://www.computerworld.com/s/article/9239513/Could_the_Bitcoin_network_be_used_as_an_ ultrasecure_Notary_service_.

"Proof of Existence." Proof of Existence. Web. 27 May. 2014. http://www.proofofexistence.com/. "Virtual-Notary." Virtual-Notary. Web. May 27. 2014. http://virtual-notary.org/.

“Commitment Scheme” Web. 16 November. 2014.

http://en.wikipedia.org/wiki/Commitment_scheme

“Foundations of Cryptography: Volume 1, Basic Tools, (draft available from author’s site).” Cambridge University Press. ISBN 0-521-79172-3. 16 November. 2014. (see also http://www.wisdom.weizmann.ac.il/~oded/foc-book.html) :224

“Real-World Sybil Attacks in BitTorrent Mainline DHT Wang Liang. Jussi Kangasharju. University of Helsinki. Web. 17 Nov. 2014. http://www.cs.helsinki.fi/u/lxwang/publications/security.pdf

“Sybil-resident DHT routing” University of Cambridge. Danezis George. Chris Lesniewski-Laas. Kaashoek M. Frans. Anderson Ross. Web. 17 Nov. 2014. https://www.cl.cam.ac.uk/~rja14/Papers/sybildht.pdf

“A Sybil-proof one-drop DHT” Lesniewski-Laas Chris. Web. 17 Nov. 2014.

http://pdos.csail.mit.edu/papers/sybil-dht-socialnets08.pdf

“Art Provenance: What It Is and How to Verify It” Web. 17 Nov. 2014.

http://www.artbusiness.com/provwarn.html

“Equine Appraisal: The Value of our Horses” Web. 17 Nov. 2014.

http://www.hgexperts.com/article.asp?id=7366

“Proof of work” Web. 17 Nov. 2014.

https://en.bitcoin.it/wiki/Proof_of_work

“Why one time passwords using nested hash chain are not used” Web. 17 Nov. 2014.

http://security.stackexchange.com/questions/35135/why-one-time-passwords-using-nested- hash-chain-are-not-used

”Proving Your Bitcoin Reserves” Web. 17 Nov. 2014.

https://iwilcox.me.uk/2014/proving-bitcoin-reserves

“Distributed Consensus from Proof of Stake is Impossible” Web. 17 Nov. 2014.

https://download.wpsoftware.net/bitcoin/pos.pdf

— - —

補遺1: 監査のアプリケーションの例、現在どのように有用か？どのようにして有用なアプリケーションをFactomのプロトコルを使って作るか

‘アプリケーション’は、ユーザーサイドのソフトウェアのためにある言葉で、Factomのシステムから読み出し、書き込むアプリケーションだ。ヒューマンインターフェイスを持ち、もしくは完全に自動化されているかもしれません。アプリケーションは、アプリケーションが必要としているチェーンによって組織されているデータに興味があります。

アプリケーションは、分散形のアプリケーション(DApps)で、Factomとともに付加的なサービスを提供するかもしれません。例えば、トレーディングのエンジンのようにとても速いトランザクションのプロセスをもち、とても正確なタイムスタンピングを持つものがあります。例えばアプリケーションで、ストリーものトランザクションであるにもかかわらず、Factomのチェーンに、文書化して、台帳を安全に保つものかもしれません。そのメカニズムは、リアルタイムの暗号的なプロセスの証明、と、保管と、そのコミュニケーションを提供します。

どのようにして、安全で分散形のログのプラットフォームを実装するかを見てみます。ログの解析は複雑なタスクです。加えてログは、簡単に騙すことが出来ます。そして、異なるそれぞれのシステムが独立して生成して、様々なメディアに保存されます（ファイル、データベース、クラウド等)

Factomでは、一意にデザインされた暗号的監査のツールが、ログ解析を安全で、簡単で、より強力なものとすることが出来ます。

例を見てみよう。Bank（銀行, B）、Payment Provider(決済事業者, PP)、そしてBitcoinの会社(BC)を見てみると、それらは下記のように相互に作用します、

1. ユーザーは、BCのウェブサイトへ行き、そして幾ら かのビットコインを購入したいと思う。

2. ユーザーは、5分間有効な引用を求める。

3. そしてPPのウェブサイトに移動される

4. PPはBのプラットフォームと繋がり、ユーザーのアカウントのお金は引き落とされる。

5. BはPPを気づかせ、そしてユーザーのアカウントから引き落とされる。

6. PPはBCにメッセージを送る

7. BCはビットコインをユーザーに送る

このことは、一般的な複数の固定のレートがあるビットコインのシナリオです。しかし、幾つかの理由においてBCが決済の通知を、ユーザーがPPに対して支払った4時間後に受け取る事となってしまう場合、誰に過ちがあるだろうか？ユーザーだろうか？銀行だろうか?決済のプラットフォームだろうか？

もし似たような決済の問題は何百何千もの決済に対して、問題が同定されて解決される前に、何日、何週間もの時間に渡って決済の問題が起こった時にどのようにして解決すればよいだろうか？誰が、”立証可能で”それらの損失/損害をもたらしてしまうのだろうか？

現在の技術的な手動の監査の仕組みログを用いることに依って、必要であるだろうこと、そして、法的な認証に必要かもしれないこと。Factomと正しい監査のアプリケーションを用いることに依って、何処から問題が生じたかを発見することが簡単になり、記録を行うことが不可能なのは時38代遅れの問題となる。
基本的に、あらゆるシステムでは、セキュアなブロードキャストのチャンネルにおいて、リアルタイムで公開される関連する証跡を出版する。
ここに他のFactomがどのようにしてビットコインの取引所の監査証跡として役に立つかについての例を記します。
それは、支払い能力の証明のメソッドをビットコインの取引所の監査に使い、育て重要なトレンドとすることだ。しかしながら、重要な脆弱性は、Factomのセキュアなブロードキャストのチャンネルを適切に使わなければ解決出来ないという問題があります。

マークルツリーによる、弁済能力の証明の方法は、Maxwell-Todd proposal,があり、ユーザーは、手動で自らの収支を報告しなければなりません。そして適切に金融機関の宣言の傾きに対して組み込まれなければなりません。提案された解決策は、十分なユーザーがもしアカウントがツリーに含まれていることを証明したならば、解決策は上手くワークします、そしてアカウントが含まれていなければ、その状態は報告されます。1つの潜在的なリスクは、データベースの所有者が、本当のデータベースを全く表していないハッシュを生成することが出来る事です。このことによって、取引所は、顧客への保証の責任を小さくした不完全なデータベースをハッシュして、あたかも弁済能力があるかのように見せかけるかもしれないことです。

ここにその詐欺の取引所が簡単にアカウントを除外することが出来ることについてのシナリオを記載します。

* Colluding Whales 攻撃:

大きなビットコインのトレーダーが、様々な取引所で行動し、市場を大きく動かしていることの署名である。そのようなトレーダーは、キャピタルの保

証を最も大きな取引所で素早くオーダーを実行する必要がある。良く、トレーダーは信頼出来るエクスチェンジを選ぶという。このようにしてハックや、流動性の問題が生じます。
彼らは自らの資金を手に入れることを第一に考えます。このような場合には、その取引所とトレーダーは、くじらのアカウントの収支をデータベースからハッシュされる前に取り除いてしまう。取引所のトップ10の大きなアカウントが、簡単に5% から20%程の取引所の信頼性を、それゆえに僅かなアカウントが不正をおこなうだけでとても大きなインパクトになってしまいます。

* “サイトの改ざん”攻撃: 現在弁済能力の証明の監査は、そうした機関のウェブサイトにおいて報告されてきた。（そのハッシュツリー）。しかし、このことは、ユーザーにとっては何の保証にもなっていない。何故なら悪意のある取引所は、異なるグループのユーザーの状態とバランスを提出すること、もしくはその状態に遡って変更することが出来るからです。それゆえ、データをFactomのセキュアなブロードキャストのチャネルを通してデータを公開することと、頻繁に公開することは根本的に重要です。

1つめの攻撃は、明らかではないものの、2つめの攻撃は、Factomを使うことに依って解決出来ます。この文書が、両替所の監査のメカニズムに対してフォーカスしていないため、我々は、詳細については立ち入らない。しかしながら基本的なコンセプトは、頻繁にタイムスタンプされた両替所のデータベースのマークルハッシュのコピーを持つことに依って、大きなバランスが勝手に含まれていないか、除外されたりしていないかどうかを発見できることがある。
そして、監査をする人は簡単に、大きな含有や、除外を手動でも発見することが出来る。覚えておいてほしいことは、トレーダーは、究極的には、両替所から何れかのタイミングで、必ず引き落としたり、入金したりする必要があるため、そのことは、銀行の履歴や、ビットコインの移管の履歴に表れるということだ。

そのような詐欺の戦略を見つけるための、伝統的な監査の産業で築かれてきたプロセスがある。しかしながらそれらは、正確で証明可能で検証可能で、改変不可能な時系列順の、追究対象に対しての情報があって初めて行うことが出来ます。

補遺2: Factomに対しての攻撃

スパムからのサービスの拒絶

Factomはオープンなシステムであるため、全てのユーザがエントリーを殆ど全てのチェーンに挿入することが出来る。

ビットコインでも同様の現象が起きている。アプリケーションがトランザクションをリジェクトするためには、そのアプリケーションは先ずダウンロードして、プロセスする必要があるだろう。そして本物ではない大量のエントリーは、最初のアプリケーションのトランザクションのプロセスを遅くすることがが出来る。この脅威は、攻撃者が費用を使わなければならないことに

よって軽減される事が出来る。このことはEmailのスパムに対するAdam BackのHash Cashの解決策と似ている。

監査はスパムに対しての他の有用なツールとなります。もしアプリケーションがセキュリティを有用性に対してトレードオフしようとするならば、監査者は、”無視する”リストをチェーン上にポストし、自らの監査のチェーンをリスト上に作る事が出来ます。監査する人は、プロフィールのチェーンを作って、他の監査者からのレビューをもらうことによる自らの名声を示すことが出来る。もし、あらゆる監査者が、悪い評判であるならば、簡単に証明可能であり、その記録は永続的なものとなる。いくつかの証明のプロセスは、オプションが異なるという点で、グレーである。実装を特定して問題を解決することが出来るでしょう。

DHTに対してのシビルアタック

一般的に分散形のハッシュテーブルは、取り分けシビルアタックに対して攻撃を受けやすい
攻撃者は多数のピアーを作って、正直なノードが通信を行うことを難しくすることが出来る。

シビルアタックは、BitTorrentのネットワークのルーティングテーブルにおいて、観察することが出来る。
これらの攻撃についてこちらの文書に記述があります。“Real-World Sybil Attacks in BitTorrent Mainline DHT”

この種の攻撃に対しての戦いは活発な議論が学術的な議論においてなされている。ある被害を抑えるための技術的な方法は、複雑なルックアップの技術を使うことに依って、シビルの中から、正直なノードを見つけることだ。それは下記に研究されている。 “Sybil-resistant DHT routing”

ソーシャルネットワークをルーティングのテーブルの中に加えて、信頼のネットワークに頼ることによって、問題を軽減できる、それは下記に研究されている。“A Sybil-proof one-hop DHT”.
http://pdos.csail.mit.edu/papers/sybil-dht-socialnets08.pdf

Factomは、最新の学術的な研究とオープンソースのリサーチに依って、DHTを安全に保ちます。

マン・イン・ミドル

最近のエントリーよりも、最初のエントリーはより重要な合意です。特権的な位置にあるネットワーク上の攻撃者は、オリジナルのコミッターとして通信を傍受して、サーバーに対するチェーンの生成のリクエストを得て、本当

のリクエストよりも先に、関連するFactomのサーバーに対してコネクションを繋ぐことが出来れば、このことを成すことが出来ます。

Factomは、チェーンがこの攻撃を防ぐために、遅延の方法を使っています。Namecoinでの解決策のように、最初にチェーンIDのために費用を支払ったものがチェーンに対しての最初のエントリーを作り出す権利があることを保証するための、関連する秘密のセットを持てるようにしています。

支払いは下記の3つのデータを含みます

*チェーンIDのハッシュ
*エントリーのチェーンID+ハッシュのハッシュ

* エントリーのハッシュ

攻撃者は、購入しようとしている事を見ることが出来るが、チェーンIDを知ることは出来ない。彼らはチェーンIDを知らないために、正しい上記の2つ目のパラメーターを作り出すことが出来ない。一旦支払いが記録されると、ユーザーはディレクトリーブロックが新しいチェーンの支払いが終わる迄待ち、次のディレクトリーブロックまで待ち、そこには1,2分の間の遅延があります。

そしてユーザーは下記を明かします。

・(Byte arrayの一連)チェーンIDとしてハッシュされるチェーンの名前

・ 支払った際に指定したエントリーと一致するエントリー

攻撃者は誤った支払いを作り出すことは出来ません、そして最初のエントリーをチェーンIDを明かすこと無く、導入することは出来ません。何故なら、早い時期の支払いに合致するため、攻撃者のトランザクションが不正であると示されるためです。

ディレクトリアタック

このケースでは、攻撃者は可能性があるか、望ましいものとして考えられるチェーンの名前を走査し、それらのIDのハッシュを生成します。そして、それらと同じChainIDを作ろうとする誰かを監視しています。

すると攻撃者と誰かのChainIDが合致した場合、攻撃者はフロントランニングを行うことが出来ます。合致したチェーンでは、彼らはチェーンのIDを知っているので、適合するが悪意があるエントリーを作り、適合するチェーンの決済を生成し、ユーザーの決済の代わりに使うことが出来ます。

もし攻撃者が、ユーザーに先立ってチェーンIDを手に入れたら彼らは勝利するでしょう。ディクショナリアタックに対しての防衛方法は、一般的な名前空間を避けることと、様々な長く続いているネットワーク上のノードに対して支払いを行うことです。Factomでは、チェーンの名前空間の定義に対しての柔軟性によって、ネームスペースの独占を非効率的なものにしています。

チェーンの拒否

このケースでは、攻撃者はアプリケーションが望むチェーンを、チェーンの仕様パターンに基づいて予測します。そしてそれらのチェーンに対して攻撃者は費用を払うけれども、決してエントリーをそこに送りません。

Factomは、そのチェーンに他のエントリーを許可する前に、支払いが行われたそのエントリーの提出を待っているため、チェーンは効率的にロックされ、決してエントリーが来なかったことが証明される迄、チェーンは効率的にロックされます。

Factomは、拒否を遅延に変換しています。10分後にもし何もエントリーが来なかった場合には、支払いは期限切れとなります。攻撃者は支払い終わった後にも、チェーンをロックし続けるためには費用を払わなければなりません。また一方で、アプリケーションは、ロックされたチェーンを飛び越えて、少し変えた名前でもって使うことで、ネーミングのプロセスを使うことが出来ます。

詐欺を行うサーバー

全てのFactomへのエントリーはユーザーからの署名、もしくはユーザーから署名されたハッシュと適合することが求められている。そしてこのことは騙す傾向のあるFederation Pool内にある、Federatedサーバーは、大変限られた攻撃しか、プロトコルの中で行うことが出来ないことを示している。

不正なエントリーは承認されず、不正なエントリーは公開されない。正直なサーバーは、直ぐにServer Fault Messageを不正を行ったサーバーに対して発信する。もし多数が不正を発見したならば、そのサーバーを除かれる。多数が不正しない限りこのプロトコルは誠実であり続ける。過ちを発見できなかったFederatedサーバーはユーザーからのサポートを失うようなリスクと、Federated Serverのプールから除かれるリスクを持っている。

Federated サーバーはエントリーの支払いの記録を遅らせる事が出来る。だが、エントリーの支払いが、分散形のFactomのノードのセットを通じて支払われたのであれば、エントリーの支払いが遅れる事は記録されます。そのため、ユーザーは、他のネットワークと比べて論理的に良くないパフォーマンスを持つサーバーをサポートしないようになるでしょう。

Federated サーバーはエントリーの記録を遅らせることが出来ます。そして支払いを極めて速く受け入れる事が出来る。だが幾つかの理由で、Federated サーバーがエントリーの記録を拒否する。そして次の瞬間には、チェーンに対しての責任は、他のサーバーへと移動します。大半のサーバーが誠実であるかぎりは、エントリーは記録されるでしょう。そして時間ごとのデータはサーバーがエントリーを遅らせようことを示します。それによって、結果的にユーザーからの支持を失うことになるでしょう。

Federated サーバーはあらゆる点において、誤りのメッセージを送ることが出来ます。他のFederated サーバーは、メッセージが理解出来なかったときに、その悪いサーバーに対してSFRを発行するでしょう。
SFRを発行する多数派のサーバーは、その悪いサーバーを起動するだろう。

そして、ネットワークはメッセージを無視して他に転送することはないでしょう。

Federated サーバーは正当なエントリーの支払いのメッセージを受け入れることを拒絶することが出来る。それは公開のアドレスが、幾つかの当事者に結び付けられているという考えのもとにだ。繰り返すと、多数派のサーバーが正直であると考えれば、その支払はコントロールが誠実なサーバーに移った時に受け入れられる。更に言えば、ノードは監視されているため、その遅れと、もしかすると遅れのパターン、そして誤った動作を行っているサーバーへはサポートが行われなくなるでしょう。

補遺3: ビットコインへのタイムスタンピング：

どのようにしてFactomのタイムスタンピングのメカニズムがブロックチェーン上のトランザクションを安全なものにするか

Factomのデータは、ビットコインのネットワークによってタイムスタンピングされ、不可逆的なものとなります。一旦ビットコインのブロックチェーン上に公開されれば、ユーザーのデータは他のビットコインのトランザクションと同様に安全なものとなる。Factomのシステムに導入されたあらゆるデータに対して手軽な証拠を得ることが可能となります。

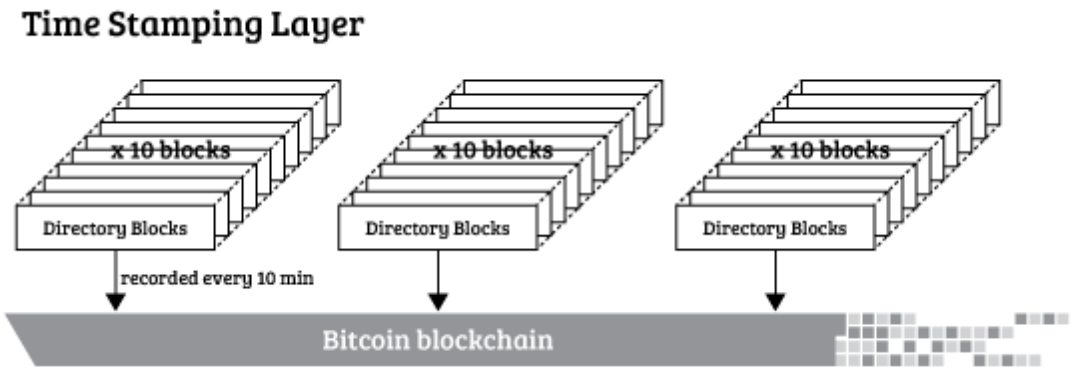
データはブロックの構造に組織され、最も高位なものがディレクトリーブロックであり、それはマークルツリーと結び付けられている。毎10分毎に、ディレクトリーブロック上のデータセットは凍結され、ビットコインのネットワークに提出される。ビットコインは、予測不可能なブロックの生成のタイミングを持つため、多かれ少なかれ、ビットコインのブロックごとに1つのFactomのタイムスタンプがあるかもしれません。

ビットコインの内部のヘッダーのブロック時間は、流動的な時間としてデザインされています。それらは、2時間現実から離れてしまうことがあります。Factomでは、標準的な時間のシステムと密着している内部のタイムスタンプを提供しています。

ユーザーのデータは、Federatedサーバーによって受け取られた時に、順序付けられます。Factomは提出されたエントリーへの参照をブロックのセットに組織します。そのFactomのためのブロック時間は1分ごとです。まとめると、Federated サーバーのネットワークは、コンセンサスを生み出し、ブロックの構造の一部となるエントリーを生成し、その時迄にタイムスタンプを押されるようになる。

一般的な注意点として、そのデータは、タイムスタンプを押される前から存在していたこととなる。
Factom上に走っているアプリケーションは、より正確で機能が豊富なタイムスタンプのシステムを、Factomに記録されようとしているよりエントリーに先立って提供することが出来ます。

Factomのタイムスタンプは、同じデータがそのFactomのタイムスタンプの後には生成されていないことだけを示します。



10のディレクトリーブロックに対してのマークルルートは、Bitcoinのブロックチェーンにトランザクションを使って挿入されます。

そのトランザクションは、OP_RETURNのアウトプットを含んでいる。それをディレクトリーブロックをアンカーリングすることとして呼んでいる。その方法では、少なくともビットコインのネットワークの様タイムスタンプのデータに対する様々な方法に対して損害を与えます。

2つのブロックチェーン内のOP_RETURNのデータに対しての可能性のある代替手段は、P2Poolのヘッダーに結びつけるか、（chronobitのように）、もしくはビットコインのブロックのヘッダーのコインベース部分に結びつけるかです。

P2Poolのヘッダーは、P2Poolのルールを満たすブロックを見つけるための数時間のマイニングを必要とする。そして、Factomのプロトコルに複雑さを追加することには、メリットがないだろう。マークルルートをブロックのコインベースの中に含めることで、マイナーの協調を要求し、
コインベースのエントリーは、暗号的な署名をFactomのシステムから得ることを必要としています。それゆえに署名されたトランザクションに比べて殆どスペースを取らないでしょう。

しかしそれはあまり実りあるものでないだけではなく、ヘッダーのみのダウンロードに含まれることとなり、BitcoinのSPV(Simple Payment Verification)クライアントに対しても影響をあたえることとなってしまうでしょう。
最初の利用可能な40のアンカーの最初の2バイトは使用決定者のタグであり（”Fa”の2バイトを用いる)そのFactomのアンカー(32バイト)は、そのタグに結合されます。そして次にブロックの高さが追加されます。（6バイト迄で、500,000年を1分のブロックとして）。仕様決定者のタグは、トランザクションがFactomのアンカーであることを指し示します。他のクオリティの決定は求められるが、タグとFactomのブロックの高さは、そうでなければ精査されなければならないようになってしまうでしょう。OP_RETURNの大半のトランザクションをいらなくしています。

OP_RETURN中のブロックの高さは、ビットコインのブロックチェーンがアンカーを順番に記録していない場所においてリストのマークル・ルートのデータは10個のディレクトリーブロックのマークル・ルートを含んでいま

す。

アンカーされたデータは、10のディレクトリーブロックのマークル・ルーとのリストを返すでしょう。そして、ディレクトリーブロックに対してのDHTのキーを代わりに返します。

マークル・ルートのタイムスタンプは、ビットコインのブロックチェーンに、Federated サーバーの一つとして、挿入される。その委譲されたサーバーのは、Federationサーバーが集めたデータにタイムスタンプを押し、ビットコインのトランザクションを作成する。そのトランザクションは、ビットコインのネットワークにブロードキャストされ、ビットコインのブロックの中に含まれる。

ビットコインのトランザクションはFactomのアンカーのように働くが、Factomのサーバーとして知られているアドレスから使用されるのではなく、ジャンクであるか、Factomをフォーク使用としているものにみえる。多くのユーザーとアプリケーションはそのようなアンカーを気にしないでしょう。

ビットコインのブロックは、統計的なプロセスと共に生成される。そしてそのようなタイミングは予想することが出来ない。このことは、Factomのアンカーは、ビットコインのブロックチェーンと、そのタイムスタンプのメカニズムに挿入されたOP_RETURNによって、大雑把に時間制限されているだけであることを意味しています。

Factomのアンカーリングの真の価値は、他の誰もが、Factomの誤った履歴を作り出せないようにすることに有る。

ビットコインのマイナーの運がなく、もしくはFactomのトランザクションを含めることが遅れたために、10分間の特定の間に凍結されたFactomの状態の間と、Factomのアンカーがビットコインに現れる時間は異なるように出来ます。

補遺4: Factomと他のブロックチェーンテクノロジーとの比較:

どのようにFactomは、ビットコイン、サイドチェーンと違っているのか

Factomはビットコインとはかなり異なっている。実際には、既存のどの暗号通貨のプロジェクトからもかなり異なっている。

ビットコインのような暗号貨幣を、厳しく分散形のメソッドでトランザクションのバリデーションのために実装し、誰でも各々のトランザクションをバリデート出来るようにしている。何故ならそれぞれのトランザクションは、暗号学的な証明によって、認証されていて、どのトランザクションも騙されないからである。各々のトランザクションは、各々のトランザクションの署名を確かめ、そしてマイナーがそれぞれの有効なトランザクションのみを含めることによって、成立します。

ビットコインのプロトコルは、トランザクションとして完全です。言い換えれば、トランザクションを通じて、ビットコインを生成して、分散することはビットコインのプロトコルの中で完全に定義されています。トランザク

ション（特定のビットコインの動き）は、と新しいブロックの発見(これはビットコインをマイニングのフィーと、ブロックのリワードを提供するためのものとして考えられる)はビットコインのプロトコルの中に只含まれ、そしてビットコインのプロトコルの中には何も残しません。他の言葉でいえば、2100万のビットコインは、究極的には、プロトコルの中に永遠にあり続けます。
ペッグされたサイドチェーンは、実装されると、ビットコインの価値をブロックチェーンの外側に動かす機能を提供します。一方でペッグされた価値は、ブロックチェーンの中で停滞します。

サイドチェーンのプロトコルは、ブロックチェーンの外へと価値を移動させて、サイドチェーンの中へ価値を移動されることを可能にしてスケーラビリティを高めることを提案しています。

サイドチェーンの中では多くのトレードが起こり、そして暗号学的な証拠が（全てのトランザクションの間では無いにせよ）ブロックチェーンの内部に記録されうるでしょう。

これらは、ビットコインをビットコインの外での状態へと動かす事となります。この証明は、ビットコインのマイナーにとっても利用可能であるべきですが、トランザクションの一連のデータはサイドチェーンの背後に隠れてしまいます。

Factomでは、ある意味でスケーラビリティをあげようとしていますが、トランザクションの価値を高めようとしているわけではなく、BTCのトランザクションでないトランザクションをブロックチェーンの外側に動かそうとしています。

例えばトランザクションは、ドメインの名前の登録や、セキュリティカメラのログ、絵画の出処の追跡、そして歴史を文書化することによって、馬の価値を図れるようにすることなどがある。いくつかのものは全く価値を移動させないが、トランザクションが出版の証拠を組み立てようとしています。

サイドチェーンでは多くのトレードが起こりえます。暗号学的な証明（全てのトランザクションの間ではないにしても）はサイドチェーンと、Factomでは、どちらもトランザクションをビットコインの外へと動かそうとしています。

だがこの似ている結果は全く異なるメカニズムによってもたらされます。いくつかの点では、Factomは、BTCからFactoidへの変換を不可分な交換として利用するためにビットコインのサイドチェーンを統合しています。

どのようにFactomは他のブロックチェーンのテクノロジーとは異なるのか

多くの異なるグループでは、ビットコインが他の種類のトランザクションを管理するためのアプローチを成功させるための方法を探しています。例えば、馬や車の交換を電子的にビットコインの拡張を使って行おうとするものがある。更には、メタルや先物や、債権といったトレードに使われる商品で

さえも賢いエンコーディングによって、ビットコインのブロックチェーンに挿入することで行おうとしています。

ビットコインを拡大し、これらのトレードを行えるようにしようとするものは、Colored Coinや、Master coinや、Counterparty上に組み上げられます。幾つかの開発者は、自らの暗号通貨を開発者と共に組み立て、より柔軟なプロトコルでもって通貨を超えたトレードを可能にしようとしています。それは、Namecoinや、Rippleや、Ethereumや、Bitsharesや、NXT等です。

オープントランザクションでは、暗号的な署名を使っていて、署名されたレシートや、ユーザーの収支の証明、（例えばユーザーはトランザクションのヒストリーを使うこと無く、最後のレシートだけで自らのバランスを証明できる。）このような方法で、中央集権的なサーバーが、クライアントの収支を変えてしまうリスク無しに、OT（オープントランザクション）は中央集権的なサーバーを、提供しています。しかし、Factomは分散形であり、ただエントリーだけを記録します。

それゆえにFactomは、OTのルールに従わないデータでも保存することが出来ます。しかしFactomはOTが行うことが出来るレートで実行することが出来ません。そうはいっても、我々は全てではないにしても、あるユーザー達は暗号学的な技術でもってOTのような仕組みを、彼らの記録の中に取り入れるのではないかと期待している。

ビットコインの上にプラットフォームを建てようとするではなく、独立した独自のプラットフォームを利用するのは、柔軟性という大きなアドバンテージがある。ビットコインのプロトコルは、恣意的なデータを記録するのに向いていない。そしてbookkeepingが、ビットコインのタイプでないトランザクションに対しては必要となってしまい、このオペレーションは、ビットコインに必ずしもサポートされていません。更に言えば、ビットコインのコンセンサスの方法に基づくProof of workは、”大は小を兼ねる“解決方法ではありません。有るトランザクションが与えられた場合には、10分よりも非常に速く解決することが出来る。Rippleや、OpenTransactionでは、コンセンサスの方法を変更することに依って、承認時間を大変にスピードアップしました。

Factomの上に立つアプリケーションは、アセットをトラックし、契約を実装する能力を得ようとし、ブロックチェーンをダイレクトに使いやすいものとするだろう。トランザクションをブロックチェーンの中に入れ、（多く言われている”ブロックチェーンの肥大化”を参照）、Factomはそのエントリーを自らの構造の中に記録する。ベースのレベルでは、Factomはどのチェーンが、ディレクトリーブロックタイムの中で、Factomに追加されたエントリーを持っているかを記録している。これらのレコードをスキャンすることに依って、アプリケーションは、チェーンから興味のあるデータを取り出すことが出来る。Factomは各々のチェーンを独立して記録し、それゆえにアプリケーションは、欲しいと思うチェーンのデータを引っ張ることが出来る。

Factomは、ユーザーのチェーン間でのコネクションを最小にする方法で、構成されています。Factomのチェーンは、他の関係のないチェーンの中に含まれているどの情報も必要とせずバリデーションを行うことが出来ます。

補遺5 Proof of Stakeに似ている機能

Factomのコンセンサスが、Proof of stakeと似ている点と異なる点

Factomのメカニズムの中でそのポリシーと報酬のメカニズムは、Proof of stakeに似ています。Factomは大半のPoSのシステムとは違い、ユーザーのステークの補助的なセットのみを認めています。

Factomのシステムにコミットされた価値だけが、投票権のシェアをもっています。移転可能なFactoidの価値は投票のシェアを持っておらず、エントリークレジットとして変換された、移管不可能な価値のみが、投票権となる。その投票権によってFederated サーバーを選出する。
この仕組に依って、ただサービスを使うかもしれないユーザーではなく、アクティブにサービスを使っているユーザーに対して、サーバーが応答できる用になっている。

個々のユーザーは、サーバーの投票権を移譲する事が出来る。そして最大の投票権を得たFederated サーバーが、コンセンサスに至るための責任を持つ。
ビットコインの深い知識を持っているひとは、純粋なPoSのメカニズムは根本的に問題があることを知っていると思う。
2つのアタックがPoSをワークしないシステムとする。Stake Grindingと呼ばれる攻撃と、Nothing At Stakeと呼ばれる攻撃だ。FactomはPoSの要素を持っているとはいえ、これらの問題による被害は受けない。

ステークグラインディング（Stake Grinding）

Stake Grinding は、マジョリティのシェアではないが、、例えば10%のサイズのシェアを持っている攻撃者が存在する時に起こりうえます。

彼らは履歴上のブロックチェーンを費用なくフォークすることが出来ます。彼らのステークが続くブロックを作るために、常に選ばれるようにして、過去のトランザクションを並び替えることが出来ます。そのため、コストを掛けずにブロックチェーンのフォークを行うことが出来ます。そして、過去のトランザクションを再構成して、ステークを常に選ばれるようにし、結果となるブロックを生成出来るようになっています。

ビットコインは情報のドメイン、コンピューターが決定するところと、熱力学的なドメイン、人間がエネルギーが燃やすところとを強く結びつけたためにこの問題を解決しました。
熱力学的なドメインではかなりの量のリソースが使われることとなり、情報のドメインとして立証可能なようになりました。そのため、ビットコインは、誤った履歴を形成するのが非常に高価す。

Factomでは、代わりの履歴を結果の後に作ることは出来ない。なぜなら、トランザクションをビットコインのブロックの歴史のブロックに入れることが出来ないためです。そのため、並行した履歴を監査されることなしに、作ることは出来ません。なぜなら、Factomはビットコインとビットコインのよく知られた秘密鍵と結びついているからです。

Nothing at stake

Nothing at stakeの問題は一層名状しがたいことです。ビットコインのポリシーに従わなければ、マイナーは1つのポリシーか、他のポリシーを選ばなければなりません。もし多数派に反対することを選ぶのであれば、再度コストを取り戻す機会もなしに、多くの電気を食わなければなりません。

PoSのマイナーはこのジレンマに陥らずに済みます。彼らは、コストを掛けずに、どちらのポリシーにも適応するようなフォークを作って、掛けのリスクをヘッジすることが出来ます。

マイナーは同時にどちらの否定にも肯定する事が出来ます、そのことによって、ダブルスペンドの攻撃が出来る経済システムとなってしまっています。2人の商売人のうちの1人が、異なるフォークを追いかけていたがために、ついには、そのお金は価値のなくなるものになるかもしれません。

ビットコインは、智性をもたず、不明瞭な点がない正しいフォークを選び出すルールによって問題を解決しました。正しいフォークは最も大きなProof of workと結びついているフォークとなります。Factomも同様に智性を持たず、不明瞭な点が無い自動的な正しいフォークを選び出すルールを、今後持つことになるでしょう。そして現在、そのようなフォークの選出のルールが今後出来上がろうとしています。