なぜDecentralized Identifiers(DID)を使う必要はないのか

公開: 2025/05/12 | Zenn (idroit-community)

本記事では、DIDを技術選定する難しさを説明する上で、その理由について言及します。 技術に批判的な立場を明言するために、若干攻撃的なタイトルをつけています。 理由に入る前に、そもそものDIDというネーミングの問題点に関して触れたいと思います。

前提: ミスリーディングなネーミング

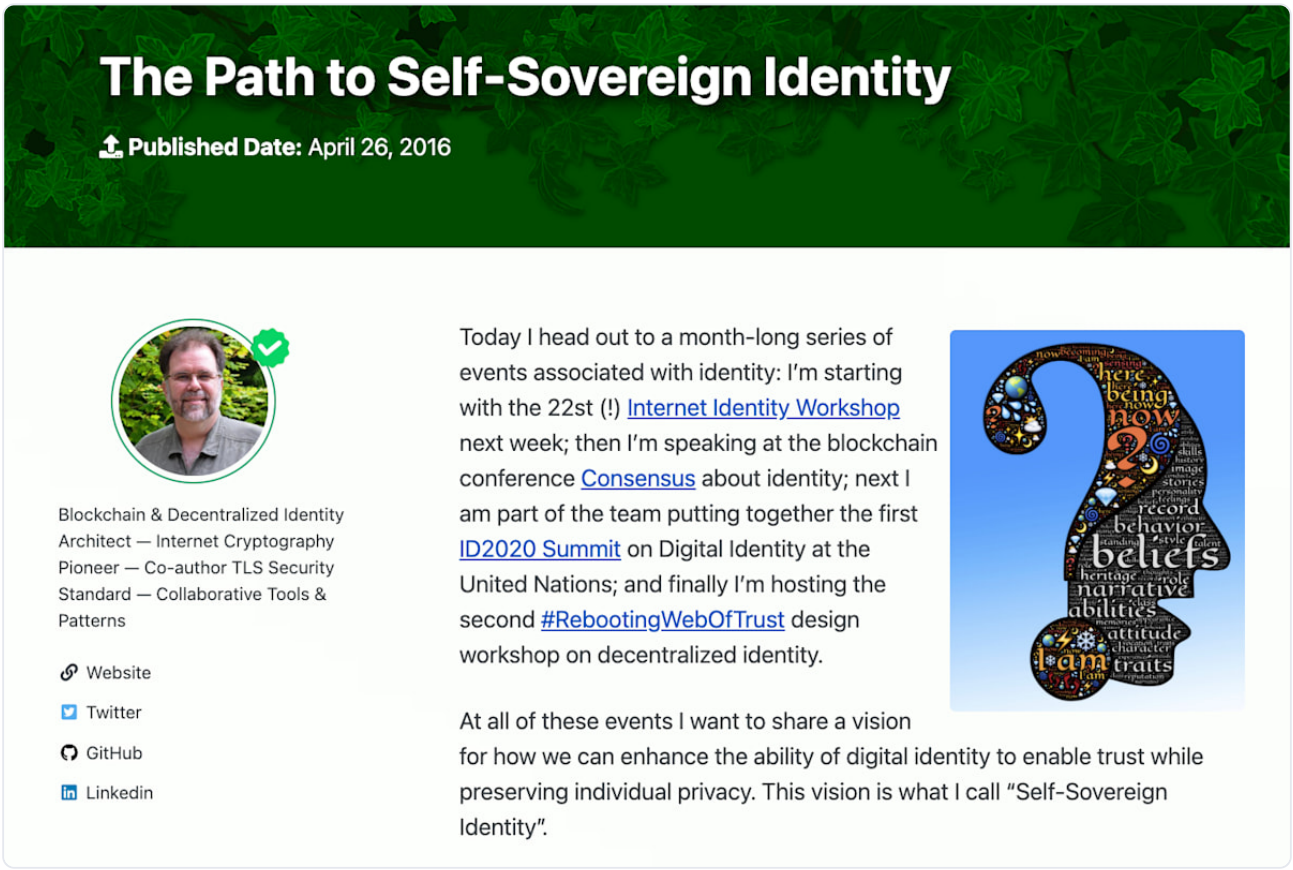
DIDはその名の通り、Decentralized Identifiers、分散型IDという名前で呼ばれています。これは**技術に対して過度に期待を煽るWeb3寄りなネーミングをしているという点で、ミスリーディングなネーミングである**と筆者は感じています。実際のところ、**DIDは分散型のIdentifierではありません**。

この名前が付けられた背景について説明します(あくまでこれは私の解釈です)。

DID登場の背景

DID登場以前、Google, FacebookなどのIdentity Provider(IdP)への依存が問題視されていました。

そんな中、Christopher Allenが「The Path of Self-Sovereign Identity」という記事にてSelf-Sovereign Identity(SSI)、いわゆる自己主権型のアイデンティティが一気に注目され始めました。Christopher AllenはTLSを提案した論文の共著者でもあります。(SSIについては詳しく別の記事にて議論予定です)



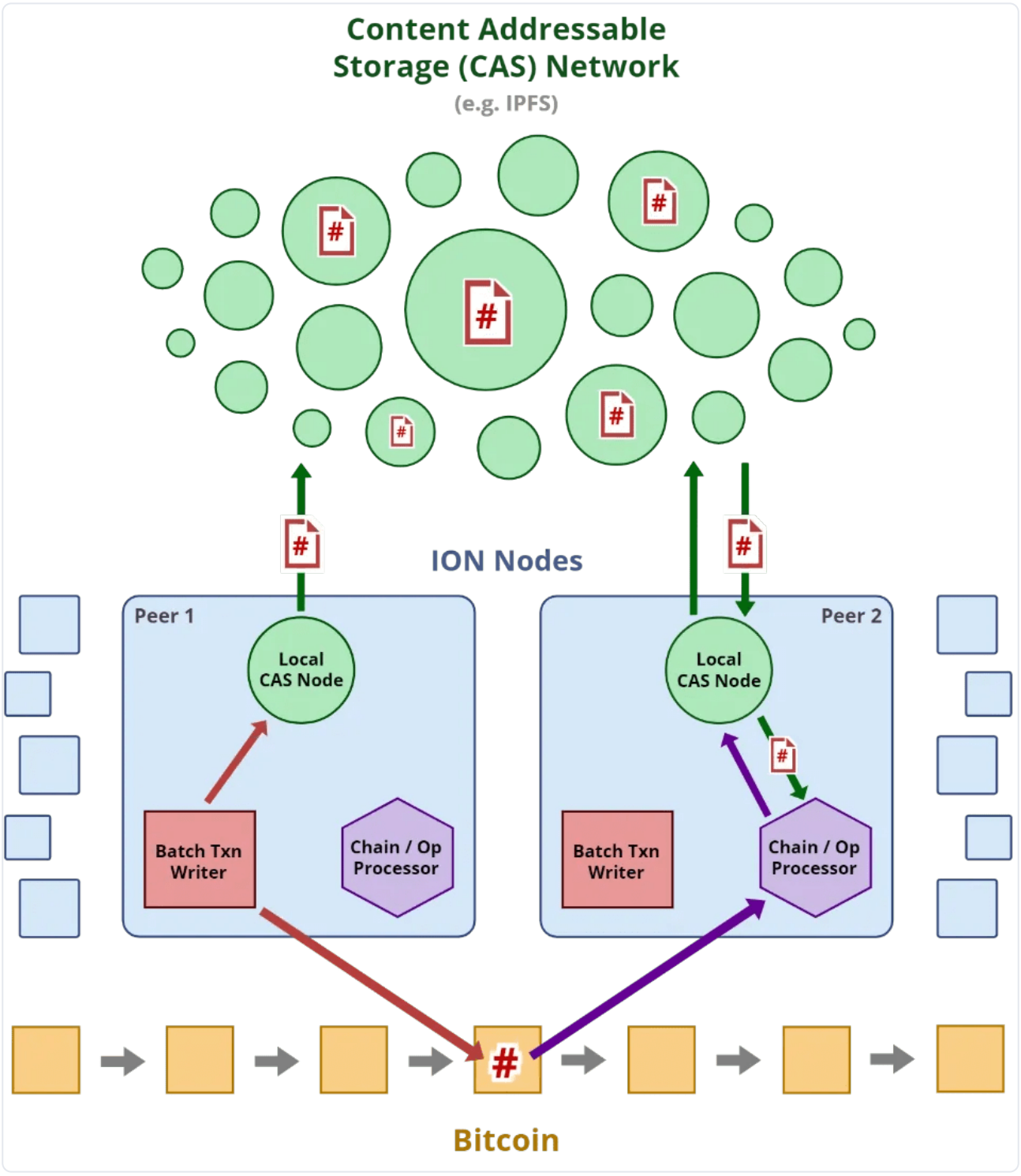
The Path of Self-Sovereign Identity（スクリーンショット）

また、彼はBitcoinコミュニティに初期から関わっていた人物です。Bitcoinの非中央集権的で暗号技術をベースとした性質が、彼が提唱するSSIにおいて重要な影響を与えたことが推察できます。そんな**ブロックチェーンを軸において生まれた技術なため、DIDという名前がついているのだと思います。** ただ**現在DIDにおいてブロックチェーンに依存しないDIDメソッドが多くあります。また、近年多くの実証実験で採用されているDIDメソッドはdid:webやdid:key、did:jwkなどの、決して分散化されておらず、分権的でもありません。** Decentralizedという用語がブロックチェーンに依存し、かつ過度に期待を煽るようなネーミングになってしまっていると筆者は感じています。

では、本筋であるDIDの技術選定が困難な理由に入っていこうと思います。

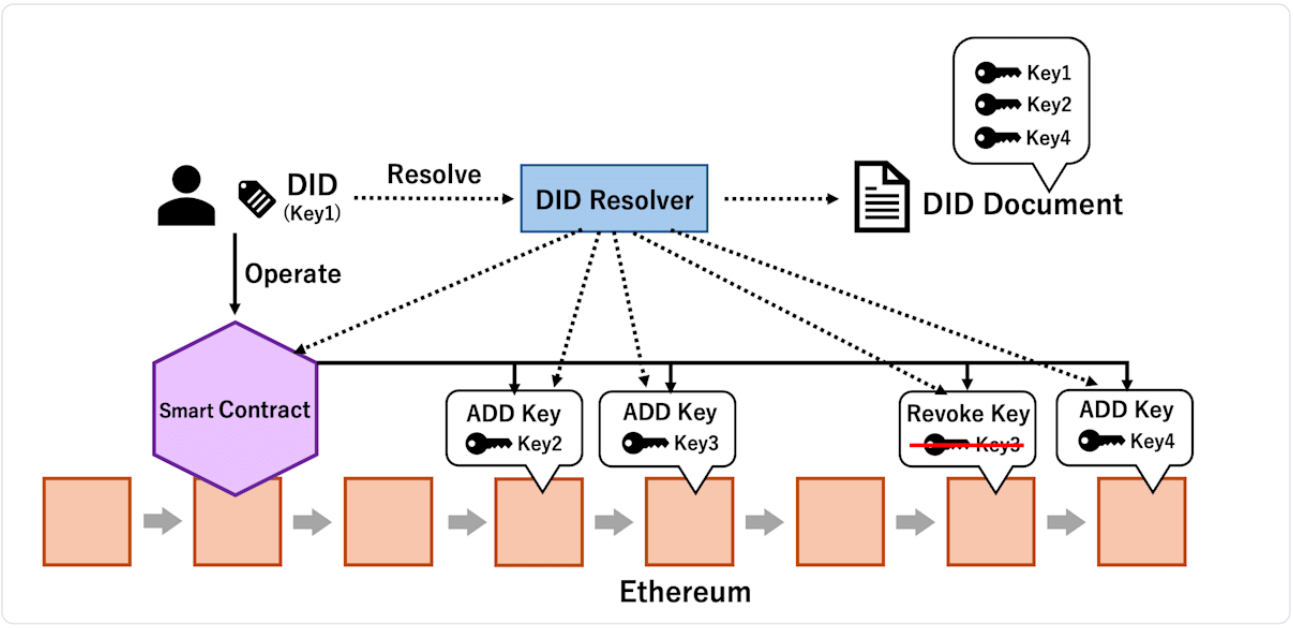
理由1: パブリックブロックチェーン依存のDIDにおける手数料

この項目においては**パブリックブロックチェーン依存のDIDのみに焦点**を当てて言及します。



did:ion におけるBitcoinブロックチェーンへのアンカリング

Sidetreeプロトコルをもとに Bitcoinブロックチェーンのレイヤー2技術として生まれた**did:ion(ION)**を例に取ります。**IONではDIDの生成時にBitcoinブロックチェーンにDIDドキュメントに関連する情報をアンカリングするため、トランザクション手数料が発生**します。手数料は決して安いものではなく、またBitcoinを所有する人にしかDIDを生成できないという点でかなり利用者が限定されます。



did:ethr メソッドの仕組み

一方で**Ethereumブロックチェーンをもとにしたdid:ethrメソッドではDIDの生成にあたって手数料が発生しない方法**があります。方法としては単純に`did:ethr:`の後にEthereumアドレスをそのまま続けます。Ethereumアドレスは公開鍵をもとに生成する情報であり、ブロックチェーンに対するトランザクションは発生しません。`did:ethr`メソッドでは、ブロックチェーン上にDIDに紐づく変更情報がない場合はEthereumアドレスをDIDとして使用します。鍵の失効やローテーションが発生した場合、ブロックチェーンにその旨のトランザクションを送信し、変更記録を参照可能にします(トランザクション手数料が発生)。

did:keyやdid:peerなどの鍵のローテーションや失効の仕組みのないDIDメソッドを選ぶか、はたまた将来的に変更が起きうることを想定し、あらかじめdid:ethrを選ぶかはトレードオフが生じます。

理由2: DIDを介する意味がない

この項目においては近年期待・採用されることの多い**did:web**について言及します。(did:jwk、did:dnsを含む)

did:webなどのDIDメソッドは、Issuerの公開鍵を取得する場合などにおいて重宝されることが多いと思います。自身でWebサイトを運用しているIssuerにとっては公開鍵の置き場所を明示できる便利な方法だと思います。

しかし、例えば技術要件が「識別子をもとにデジタル署名検証に用いる公開鍵を参照可能にしたい」だった場合、本当にDIDは必要なのでしょうか? この場合、公開鍵の置き方・参照方法についての取り決めがあれば十分なはずです。 以下にDIDと同じ手法が取れる標準や技術を紹介します。

JWKS(JSON Web Key Set)

JWKSはdid:webと同じように、Webサイトの特定のパス(例: https://example.com/.well-known/jwks.json)にJWK形式の公開鍵を置く方法です。

例) `jwks.json` の中

```
{
  "keys": [
    {
      "kty": "EC",
      "crv": "P-256",
      "x": "...",
      "y": "...",
      "use": "sig",
      "alg": "ES256",
      "kid": "key1"
    },
    {
      "kty": "RSA",
      "n": "...",
      "e": "...",
      "use": "sig",
      "alg": "RS256",
      "kid": "key2"
    }
  ]
}
```

JWKSをDID形式に変換した `did:jwk` というものもあります。

DNS

より単純化するならば、DNSで十分なはずです。DNSのTXTレコードに公開鍵を入れるなどで、署名検証用の公開鍵情報は取得できます。

```
selector._domainkey.example.com. IN TXT "v=DKIM1; k=rsa; p=MIIBIjANBgkqhkiG9..."
```

これに似た手法をDIDとして扱う手法が `did:dns` 、 `did:dns-sec` メソッドです。

- 例) did:dns:example.com → example.com のDNS TXTレコードを解釈して didDocument を構築

DIDを選ぶ意義?

DIDを選んだ場合、**わざわざ一度DIDリゾルバーを介してDIDを解決して公開鍵を取得するという一手間が発生します。** そもそもDIDリゾルバー機構を自身で運用する主体はどれぐらい存在するのでしょうか? 第三者のDIDリゾルバー機構を利用する場合、その第三者への依存はDIDを選定するに勝るほどの意義があるのでしょうか?

もちろんDIDだからと言って必ずしもDIDリゾルバを利用しなければいけないというわけでは決してありません。あえてDIDの形式を取ることで、VCの署名検証に用いる公開鍵であるということが把握しやすくなるといった意味合いもあるかもしれません。(決してそのような共通認識やプロトコルがあるとは思いませんが。)

理由3: ホルダーバインディング(Holder Bindings)が困難

DIDに期待する多くの人が見逃しがちなのが、**DIDで証明できる情報はProof of Possession(PoP、所有の証明)だけなのです。**

PoPはSSIなどの文脈においては、自己主権性の主張方法の一つとして役に立つのかもしれません。**しかし、識別子と実在する人物との関係性を主張する上では全くもって不十分です。**識別子と実在する人物の結びつき関係のことを、**ホルダー・バインディング(Holder Bindings)** と呼びます。 デジタルアイデンティティを主張する多くの文脈においても、DIDだけでは識別子とデジタルアイデンティティの関係性の証明にはなりません。事前に提示されたDIDは、秘密鍵を借りた、もしくは不正に奪った第三者が操作している可能性を拭いきれないからです。

DID(DID Document)にHolderの本人確認に用いる情報を付け加えることでホルダー・バインディングの問題を解決しようとする「Identifier Binding」と呼ばれる議論が過去にありました。

Identifier Binding

Identifier BindingはRebooting the Web of Trust(RWOT)コミュニティなどが中心となり、ホルダー・バインディングの問題解決に向け議論をしていました。

- リンク: https://github.com/WebOfTrustInfo/rwot11-the-hague/blob/master/final-documents/identifier-binding.pdf

端的に内容を説明すると、Holderの本人確認に使用可能ないくつかの情報をVCに含めます。例えば、下の図のように配列の一番目の要素にHolderのDID、二番目にはパスポートなどの公的身分証の情報、ポートレート画像のバイナリデータを含めます。

```
...
    "binding": [ {
       "id": [ "somevaliduri" ],
       "type": "didAuthenticationKey",
       "keyId": "did:example:deadbeefcafe#keys-3"
    }, {
       "id": [ "sbmevaliduri", "anothervaliduri" ],
       "type": "passport",
       "nationality": "NL",
       "passportNr": 012345678,
       "contentHash": "3338be69 ... 2398f392"
    }, {
       "id": [ "somevaliduri" ],
       "type": "portrait",
       "format" : "png",
       "portrait": "iVBORw0KGgoAAAANSUhEUgAAA8sAAAJl..."
    } ],
...
```

Identifier Binding の構造例

検証者はそのVCの本人確認情報をもとに、ユースケースに応じた本人確認を行います。 Identifier Bindingは一時の議論でストップし、現在は下火となっています。

問題4: DIDメソッドの乱立と相互運用性の欠如

DIDメソッドは2025年5月現在200種類以上あります。 参考: https://www.w3.org/TR/did-extensions-methods/

このようなDIDメソッドの乱立により、DIDメソッドに応じて参照先となるレジストリが増加し、DIDに対応したシステムの構築は困難です。 また、それぞれの**DIDメソッドの相互運用性がないことも問題**です。

これだけ乱立していれば、将来的に廃れるDIDメソッドも多くあると思います。実際にすでに廃れてしまったDIDメソッドもリストにあります。すでに廃れたDIDメソッドは**そのDIDに対応したリゾルバソフトウェアのメンテナンスが終了することにより、利用できなくなるという可能性**もあります。 そういう意味では**DIDにおける識別子の永続性に対して疑問視すべき**だと思います。

期待できる可能性

1. 既存のIDシステムの刷新

現在のIDインフラが全てDIDに変更された場合、DIDの参入障壁は下がります。 例えば、既存のシステムのアカウント情報や個々のデータがDIDで識別され、署名つきデータが交換されるなどの状況になることです。

そのような状況を作るには、**単一の事業者が話題性・ビジネス目的でDIDを利用するのではなく、多くの事業者や組織が連携してDIDをもとにしたID基盤を構築することが重要です。**

2. ホルダーバインディングへの新たなアプローチ

ホルダー・バインディングの問題を解決しようとする動きは健在です。

2-1. DIDと個人の関係性にお墨付きする第三者サービス

第三者サービスが識別子と本人確認情報をお墨付きする方法です。検証者はKYC業者をトラストした上で、そのDIDを特定の人物のものであると信じ、検証に用いる方法です。 (この場合におけるトラストは、IdPへの依存回避に勝るほど重要なのか、また結局IdPへの依存構造に帰着するのではないかは慎重に検討が必要だと感じます。)

2-2. 生体情報との結び付け

現在、FIDOアライアンスなどの多くの組織で生体情報と識別子やVCとの紐付け方法の議論がされています。生体情報と識別子情報がなんらかの技術やガバナンスによりバインディングされることで、解決されることがあるかもしれません。

3. 新たなユースケース創造

DIDの特性が有意義に発揮される可能性がある、私なりに考えた特定のユースケースを紹介してみます。

モノのアイデンティティ

モノのアイデンティティと呼ばれる文脈においてDIDの活用は期待されています。 IoTデバイスなどの識別にDIDを用い、センシングデータに署名を付与することでIoTデータの真正性担保などのセキュリティ向上につながる可能性があります。 もしくは、個々の商品にDIDを記載したタグを添付し、流通経路における管理責任者の所在管理としてDIDが活用できるかもしれません。

匿名性の高いコミュニティ

昨今のインターネットのトレンドとして、誹謗中傷や誤情報の拡散が問題となり、開示請求で発信者の責任を追求する方向にあると思います。日本政府もSNSを監視・規制する方向に進んでいます。 私個人としては**責任あるインターネット利用において、発信者の責任追求の手段として匿名性を無くすことだけが解決策だとは思っていません。** なんらかのトラブルが発生した場合において、匿名性を維持しつつ紛争解決の手段としてDIDを活用できるかもしれないと考えています。

終わりに

文章を通してDIDに対して批判的になってしまいましたが、私個人はDIDの未来に期待をしている人間の1人です。 私の主張としては、「期待できる可能性」で書いたような**DIDが活かされるユースケースにおいて、あくまで用法用量を守ったDIDの使い方をすべき**だと考えています。

一方で、Why DID? のクエスチョンを慎重にとらえ、正しく技術を利用することが、複雑性を生まず無駄のない方法だと筆者は考えます。 **研究を続けることで技術的に進歩が生まれ、課題を乗り越えた上でDIDのようなIDシステムが使われることを期待します。**

最後に改めて書いておきますが、この記事でまとめたことはあくまで私の解釈で、間違っている可能性は大いにあります。ご指摘お待ちしております。

出典: https://zenn.dev/idroit/articles/5d8541fc167c19