APTOS

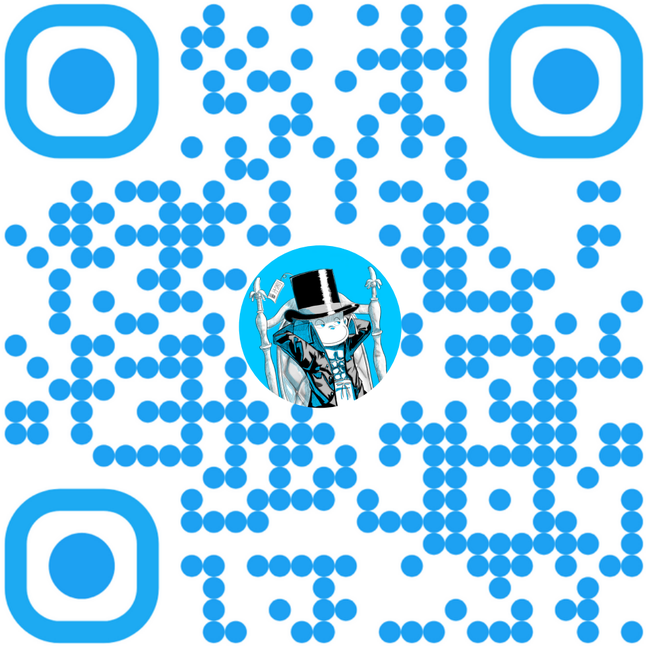
【オンライン勉強会】総額$35,000 APTOS WAVEHACKの攻略方法とは？

The Move Language: Technical Advantages

LEARN ONLY WITH SLIDES

By Yusei White

About me



Software Engineer - Cryptography @ Monas

Research:

- **Permissionless Blockchain (e.g., Consensus, smart contracts)**
- **Privacy Engineering (e.g., Applied Cryptography)**
- **Software Engineering (e.g., Distributed System, Network)**

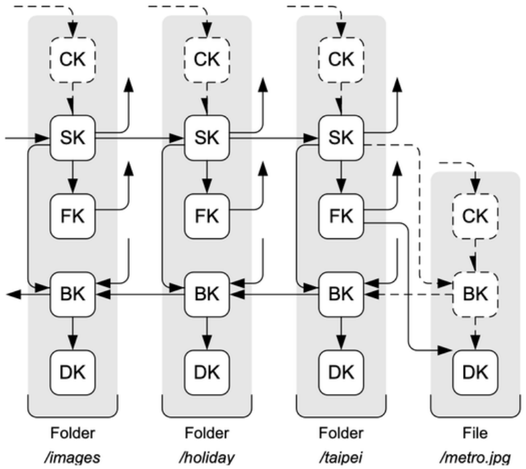
Experiences of Move:

- **Participated in Web3 Builders x Sui Hacker House**
- **1st place at Sui Builder House Kyoto**
- **3rd place at Aptos Seoul Hack DeFi Track**
- **Adopted by Web3 Startups**
- **Lots of conferences . . .**

About Monas

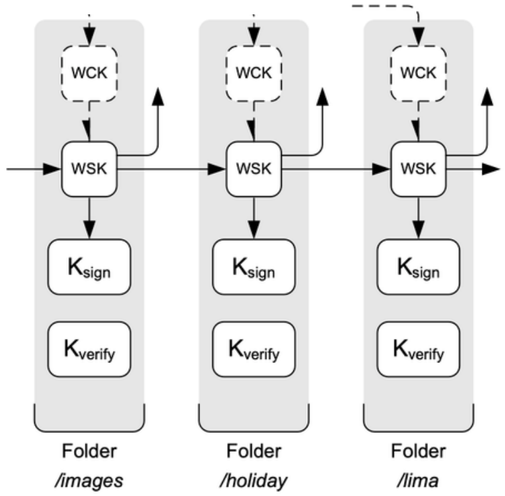
課題

- データの相互運用不可能性
- PIでさえ企業やプラットフォームが制御
- データにアクセスできない
- データのサイロ化 - データのアクセス不可能性



プロダクト

- 分散型Personal Data Store
- Cryptree: 複雑で柔軟なディレクトリ構造へのアクセス制御
- 相互運用可能なデータインフラ
- プライバシー保護
- ブロックチェーンで最新データの真正性を証明



対象者

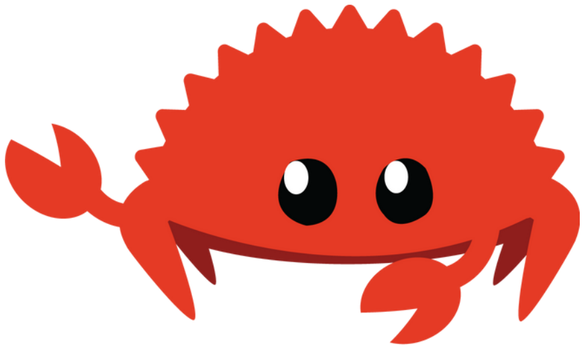
- rust知らない
- aptos知らない
- move知らない
- 技術者でない
- Ethereum聞いたことある
- スマートコントラクトとブロックチェーンについて簡単に説明できる

About Move

- 2018年開発
- FacebookのDiemプロジェクトで開発された
- Rust言語をベースとしたスマートコントラクトを記述できる言語
- FacebookにMove言語を作るチームがあった
- AptosやSuiで採用，SuiはSui Moveに改良

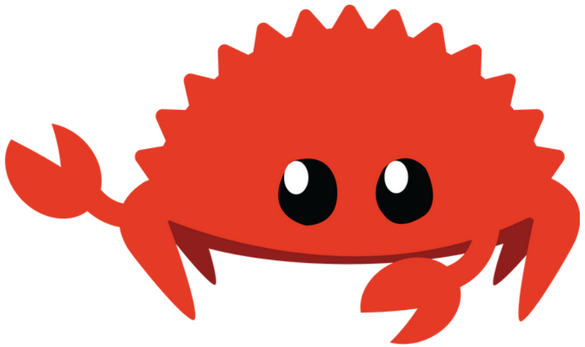
About Rust

- Move言語はこれを継承
- 現代版のC言語
    - Cの脆弱性をネイティブで対策



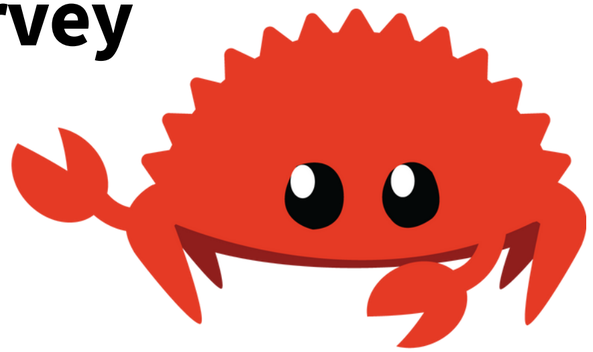
About Rust: Benefits

- 高速処理
  - 並列実行
  - 実行速度がCやC++と同程度
- メモリ安全
  - 所有権モデル <- 後述
  - ボローチェッカー
- スレッド安全
- 静的型検査
- みんな大好き <- 後述

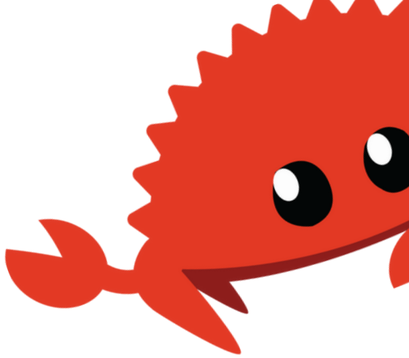


Details of Benefits

- 所有権モデル
  - 変数が必要なメモリを確保して，スコープから外れたら，値は破棄されて，メモリが解放される方式
- みんな大好き
  - 90000人以上の開発者の中からの投票により，Rustが一番好まれている
  - Stack Overflowの2023年のDeveloper Surveyというアンケートより



事例: Discord



- Rust言語活用の事例
  - Discordがパフォーマンスのために Go言語からRustへ
- 理由
  - ガベージコレクタの頻繁な実行を消したい
- 詳しく
  - ガベージコレクションしているときに，空きメモリ領域を判断するためにさまざまな処理を進める必要があり，これがプログラムの速度が低下する可能性がある

Move: Quick Question

これに答えられるようにするのがこのスライドの目的

Q. Move言語はどのようにして，ハッカーがAMMプールのオブジェクトを取り出して資金を流出させることを阻止しているのか？

Move: Quick Question

Q. Move言語はどのようにして，ハッカーがAMMプールのオブジェクトを取り出して資金を流出させることを阻止しているのか？

言い換え

- Moveでは，オブジェクトを任意のモジュールに送ることができるので，オブジェクトが信頼されていないモジュールを介するとき，どのようにオブジェクトを安全に取り扱っているのか？
- オブジェクトが信頼されていないコードによって悪用されないという保証はあるのか？

An Answer

Q. Move言語はどのようにして，ハッカーがAMMプールのオブジェクトを取り出して資金を流出させることを阻止しているのか？

A. 安全性と検証

- リソース安全 <- 後述
  - Object Model - Suiはより特徴的
  - 所有権モデル
- メモリ安全 <- Rustより継承
- 型安全 <- Rustより継承
- バイトコード検証（Bytecode Verifier）<- 後述
- 形式的検証（Move Prover) <- 後述

Move: Struct

構造体＝リソース＝デジタル資産につながる

- プリミティブ型（u8、u64、bool...）
- 他の構造体であるフィールド

リソース安全は，自分自社の資産を安全に扱えるかという最重要なこと
Moveでは，Rustを制御し，Rustの柔軟性をあえてスマートコントラクトのために低下させている
デジタルアセットのプログラミングのためにはそれなりの制限が必要ということ

Move: Struct

制御内容

1. 構造体のインスタンスをインスタンス化，破棄することは，構造体を定義しているモジュールの内部だけでしかできない
2. 構造体インスタンスのフィールドは、そのモジュールからのみ変更できる
3. 構造体のインスタンスをモジュール外にクローンまたは複製することはできない
4. 構造体インスタンスを他の構造体インスタンスのフィールドに格納もドロップもできない

Move: Struct

制御緩和策 = Ability

- Key
  - Aptosの場合
    - グローバルステートマシンに保存される
    - リソースの作成，削除，更新
  - Suiの場合
    - UID, Unique ID = オブジェクトのIDとして利用される
- Store
  - 構造体を別の構造体のフィールドとして埋め込める
- Drop
  - 構造体をどこからでも任意に破壊できる
  - これがないということは構造体はなくならないということ
- Copy
  - 構造体を任意の場所からコピー/クローンできる
  - e.g., coinだとcopyさせない = 通貨偽造ができない

Move: Bytecode Verifier

静的解析ツール

- Moveモジュールが型安全性、メモリ安全性，リソース安全性のルールを守っているかどうかをチェックする
- コンパイル時とモジュールを公開するときに使われる
- これがないと，Moveの主な利点はすべて失われる
    - スタンフォード大とfacebookの研究『Resources: A Safe Language Abstraction for Money』で示される

Move: Move Prover

スマートコントラクトの形式的検証を可能にするツール

- 可能性のあるすべての入力に対してプログラムをチェックする
- スマートコントラクトが正確か？脆弱性がないか？
  - ルール通り正しいか証明したい！

An Answer

Q. Move言語はどのようにして，ハッカーがAMMプールのオブジェクトを取り出して資金を流出させることを阻止しているのか？

A. 安全性と検証

- リソース安全
  - Object Model
  - 所有権モデル
- メモリ安全
- 型安全
- バイトコード検証（Bytecode Verifier）
- 形式的検証（Move Prover)

An Answer

開発者はロジックの実装だけに集中できる！

これで超安全かつ超高速なデジタルアセットのプログラミングを
することができる！

Performance

- Moveは一般的なバイトコード言語ではないので，ネイティブコードよりパフォーマンスが悪い
  - 必要な検証を全て行っているから
  - コントラクトの実行には問題ない SolanaもSuiもできる
  - tx処理のパフォーマンスを向上させる要因は、並列実行
    - Block-STM: 160k TPS+
      - STM = Software Transactional Memory
    - SolanaもSuiもできる

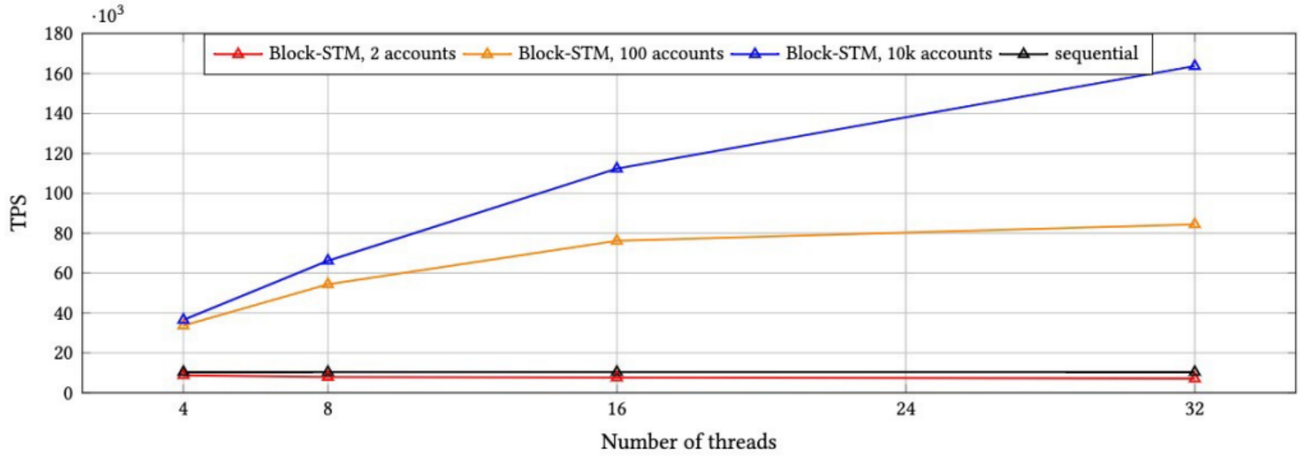
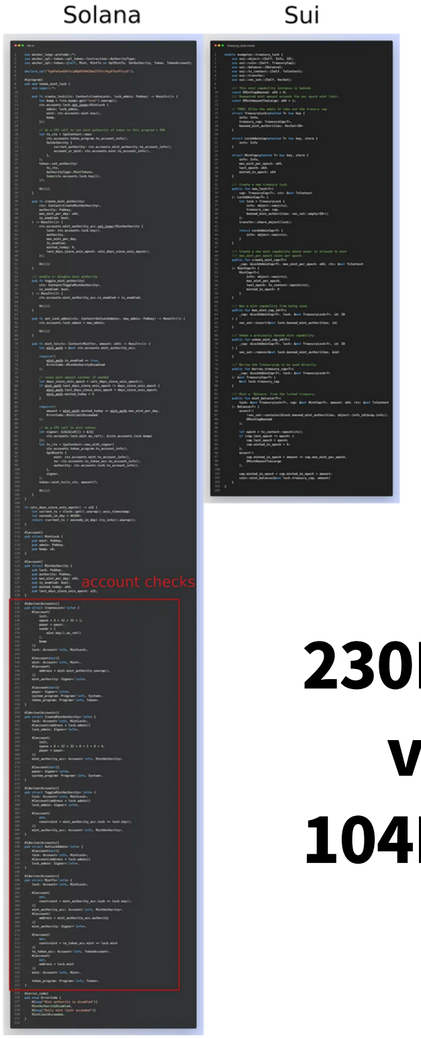


Figure 6: Block-STM (component-only) benchmarks comparing the number of physical cores with different levels of contention.

Performance

- 開発速度の向上
  - Rustの場合: 2x ~ 5x +（Solana）
  - 開発者の数を半分以下にできる
- Moveの参入障壁がRustやSolidityよりもはるかに低い
- Moveならできちゃうかも？



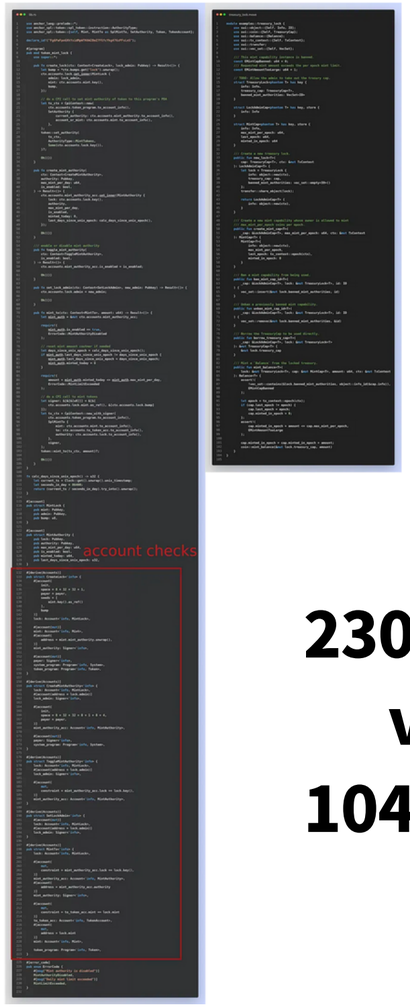
230LOC
vs
104LOC

Smart Contract Development — Move vs. Rust

Performance

Solana

Sui



230LOC
vs
104LOC

- 赤枠はSolanaのアカウントのチェック（91LOC）
  - Solanaでは，コントラクト個別でリソース安全を確保しなければならない
  - Composabilityも実装しづらい
  - Moveはしなくていい
- 不適切なアカウントチェックによるハッキング事例
  - Wormhole ($336M)
  - Cashio ($48M)
  - Crema Finance ($8.8M)

Performance

- アカウントチェックがないメリット
  - アカウントチェックを正しく実装することはめちゃくちゃ厄介
- 理由
  - 一つでもミスしたら、致命的な脆弱性やユーザー資金の損失につながる
- よって
  - 非常に機密性の高いデータのやり取りや管理もMoveならできるかも！ = Moveは難しくない！

Rust -> Move

- Move
  - Moveバイトコード自体が実行可能な表現
  - コンパイラを信頼不要
  - Move言語によって強制される保証が実行可能な表現に直接適用される
  - 信頼できないコードでも安全性が保証される

- Rust
  - ソースコードから実行可能な表現へコンパイルされる言語
  - SolanaもMove使えばいい？
    - 根本的な変更が必要でそんな簡単じゃない
    - セキュリティ懸念によるSolana上のMove VMの削除

The Market of Move

TVL, Total Value Locked

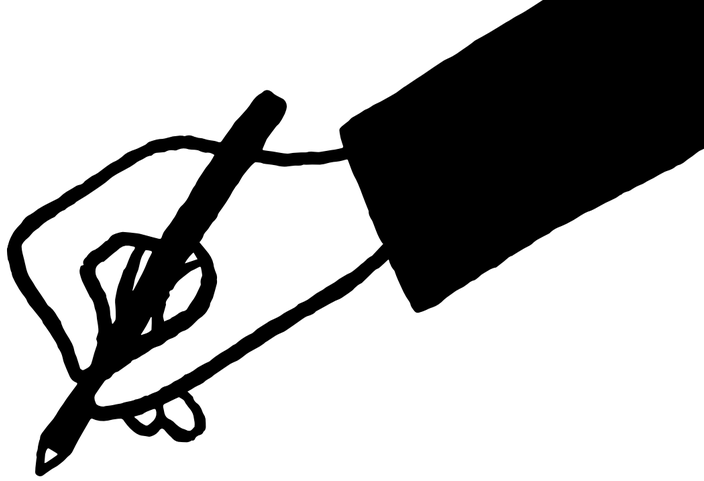
- Sui: $600M+, Non-EVMで世界2位
- Aptos: $300M+

開発者の数（月あたりのOSS開発者）

- Sui: 230人+
- Aptos: 310人+

みなさんがアーリーアダプターになることを楽しみにしています．

References:



1. Cryptree: A Folder Tree Structure for Cryptographic File System，https://raw.githubusercontent.com/ianopolous/Peergos/master/papers/wuala-cryptree.pdf
2. 実装言語を「Go」から「Rust」に変更、ゲーマー向けチャットアプリ「Discord」の課題とは，https://atmarkit.itmedia.co.jp/ait/articles/2002/10/news038.html
3. Rustプログラミング言語，https://www.rust-lang.org/ja/
4. 所有権とは？ - The Rust Programming Language 日本語，https://doc.rust-jp.rs/book-ja/ch04-01-what-is-ownership.html
5. Programming, scripting, and markup languages - 2023 Developer Survey，Stack Overflow，https://survey.stackoverflow.co/2023/#experience-years-code
6. スレッドを使用してコードを同時に走らせる - The Rust Programming Language 日本語，https://doc.rust-jp.rs/book-ja/ch16-01-threads.html
7. Object in Aptos Standards，https://aptos.dev/standards/aptos-object/
8. Abilities in Basic Concepts，https://aptos.dev/move/book/abilities/
9. Global Storage - Operators，https://aptos.dev/move/book/global-storage-operators/
10. Pull Requests #11184: Remove move_loader and librapay，https://github.com/solana-labs/solana/pull/11184
11. Resources: A Safe Language Abstraction for Money，https://arxiv.org/pdf/2004.05106.pdf
12. The Move Prover，https://www-cs.stanford.edu/~yoniz/cav20.pdf
13. Robust Safety for Move，https://arxiv.org/pdf/2110.05043.pdf
14. Aptos vs. Sui：詳細な比較，https://matometax.com/aptos-vs-sui/
15. The Aptos Blockchain: Safe, Scalable, and Upgradeable Web3 Infrastructure，https://aptos.dev/assets/files/Aptos-Whitepaper-47099b4b907b432f81fc0effd34f3b6a.pdf
16. Smart Contract Development — Move vs. Rust，https://medium.com/@kklas/smart-contract-development-move-vs-rust-4d8f84754a8f
17. DeFiLlama: Sui Total Value Locked，https://defillama.com/chain/Sui
18. DeFiLlama: Aptos Total Value Locked，https://defillama.com/chain/Aptos
19. 果物の市場規模ランキング，https://urahyoji.com/added-value-of-fruits/
20. 2023 Crypto Developer Report，Electric Capital，https://www.developerreport.com/developer-report