SBI R3 Japan

Conclave 技術紹介

アジェンダ

- 解決したい課題例
- 関連ソリューション比較
  - ✓ZKP/FHE/sMPC/Confidential Computing
- Conclaveのご紹介

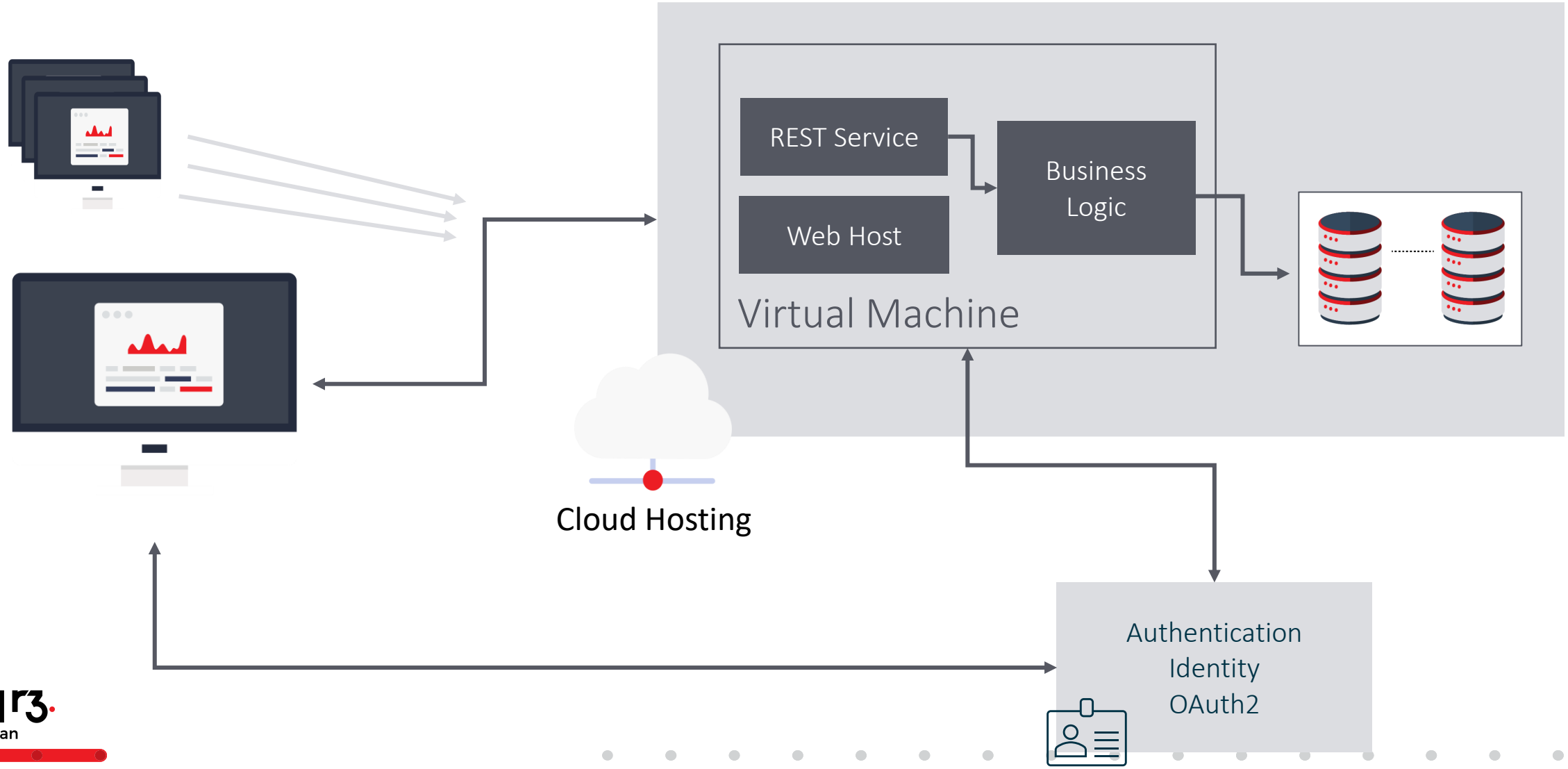
SBI R3 Japan

SBI R3 Japan

解決したい課題例

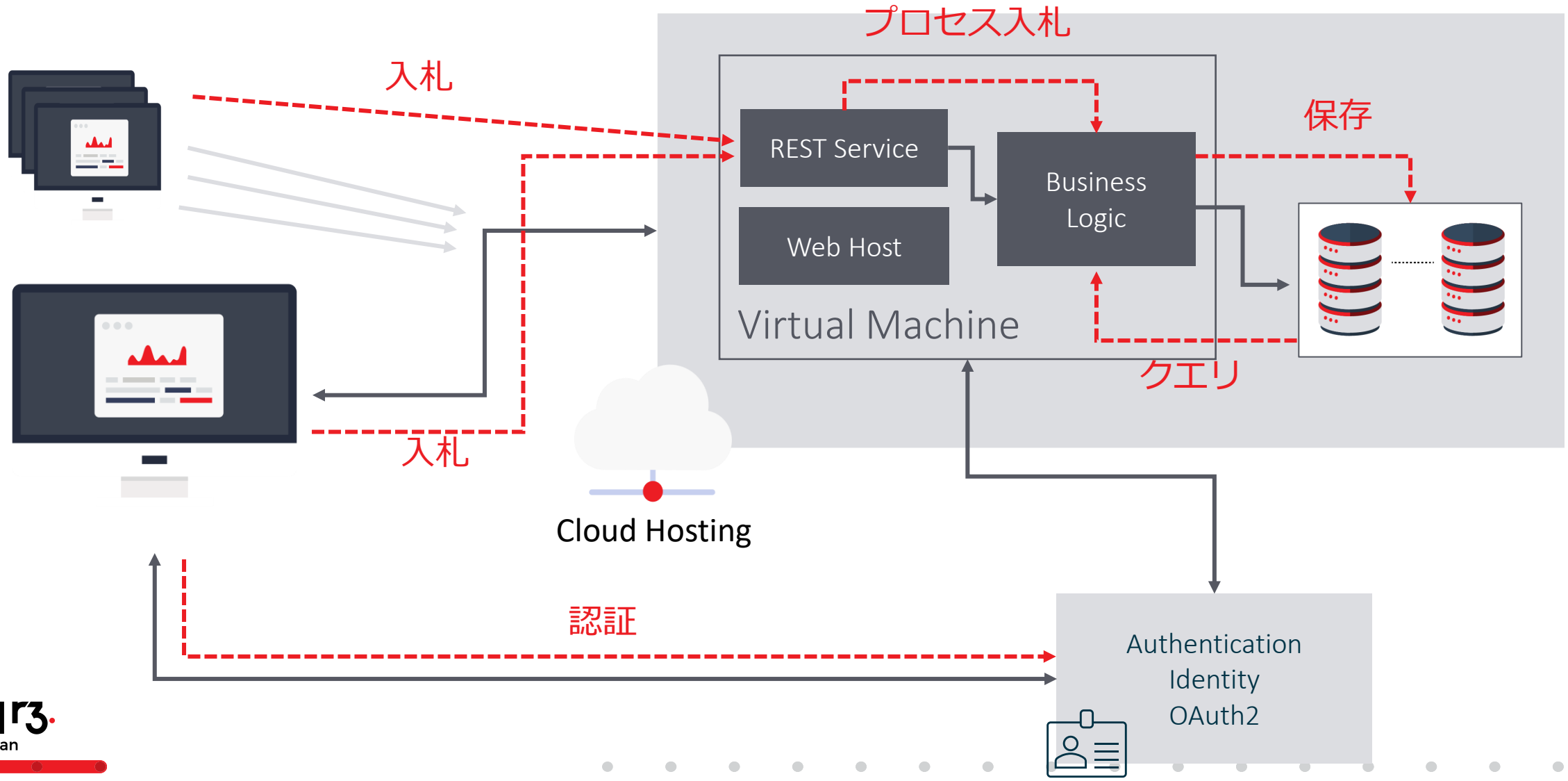
封印入札オークション

封印入札オークション　～普通のアーキテクチャ～



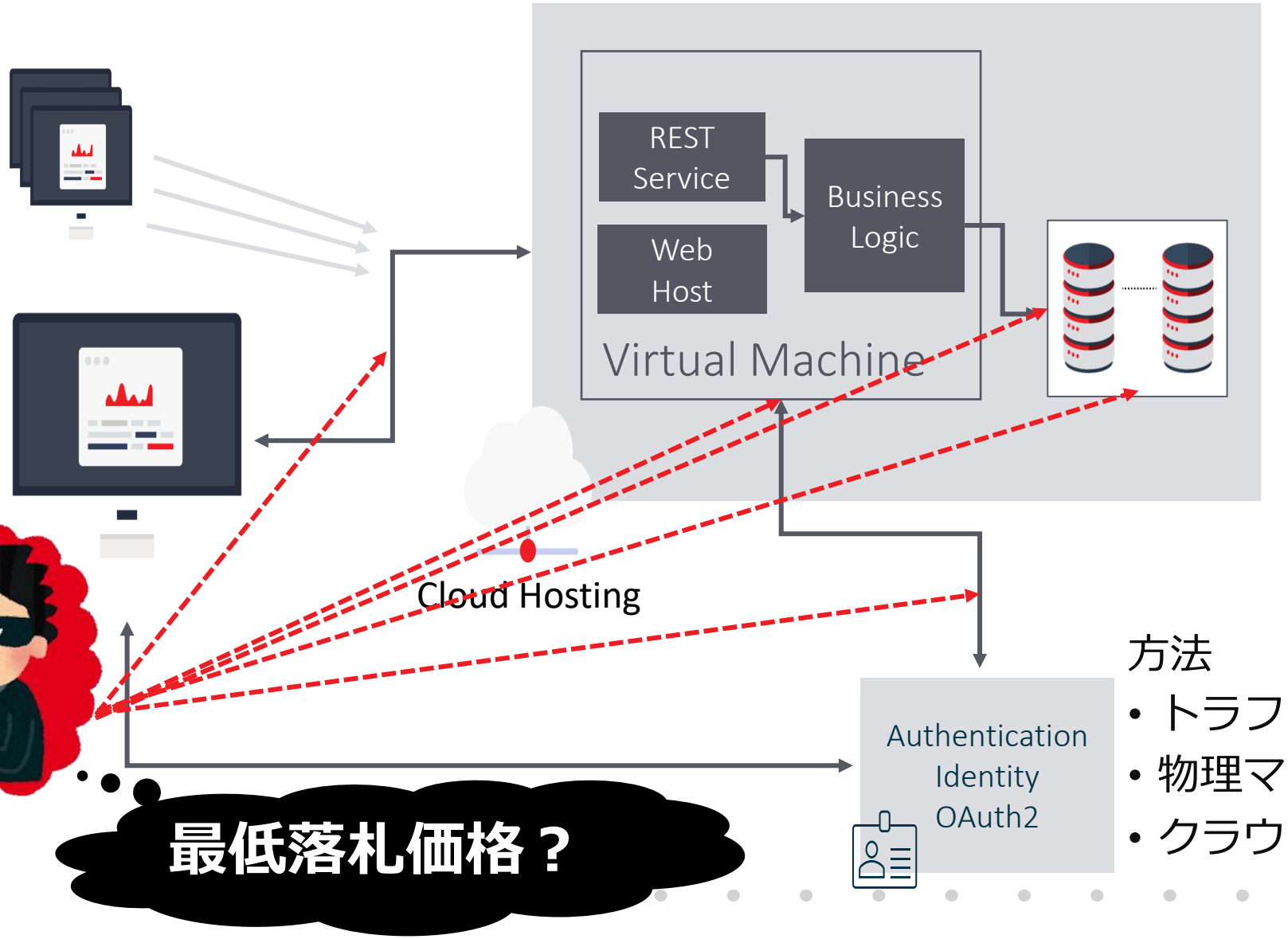
SBI R3 Japan

封印入札オークション　～プロセス～



SBI R3 Japan

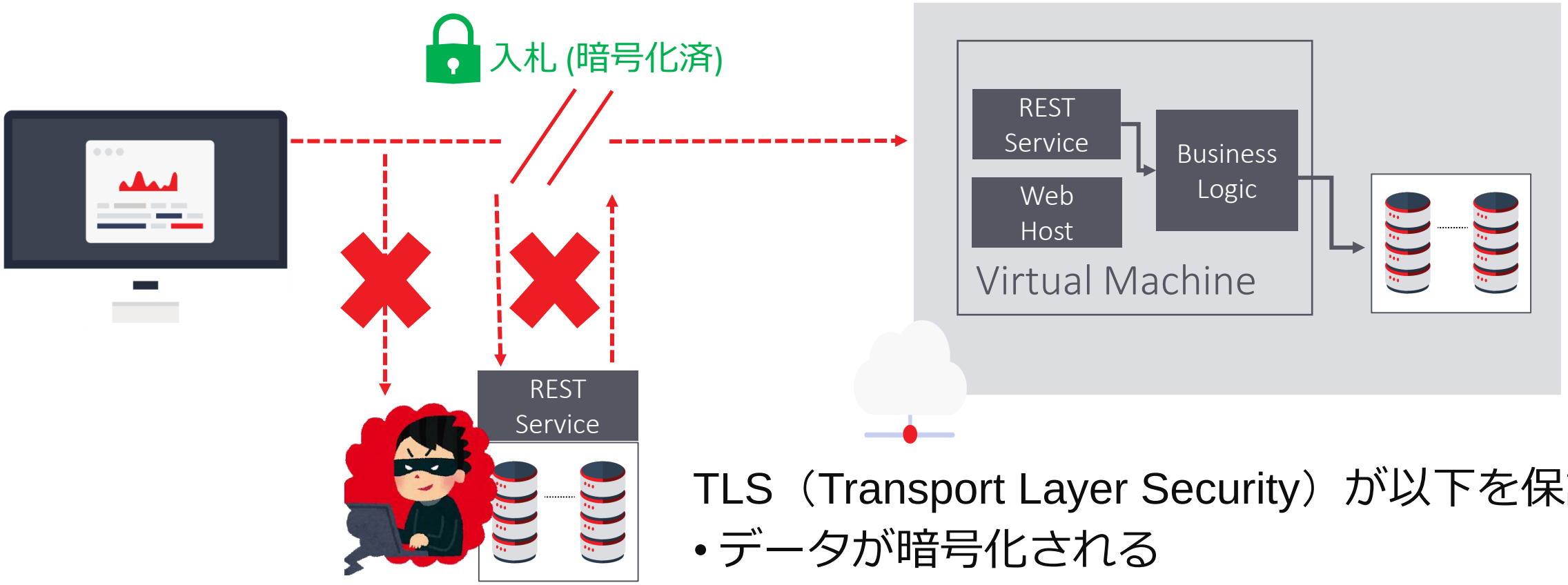
課題【悪意のある入札者・・・・】



方法
- トラフィックの傍受
- 物理マシンへのアクセス
- クラウドサービスへのアクセス

SBI R3 Japan

課題と対策①：中間攻撃者/パケット盗聴防止

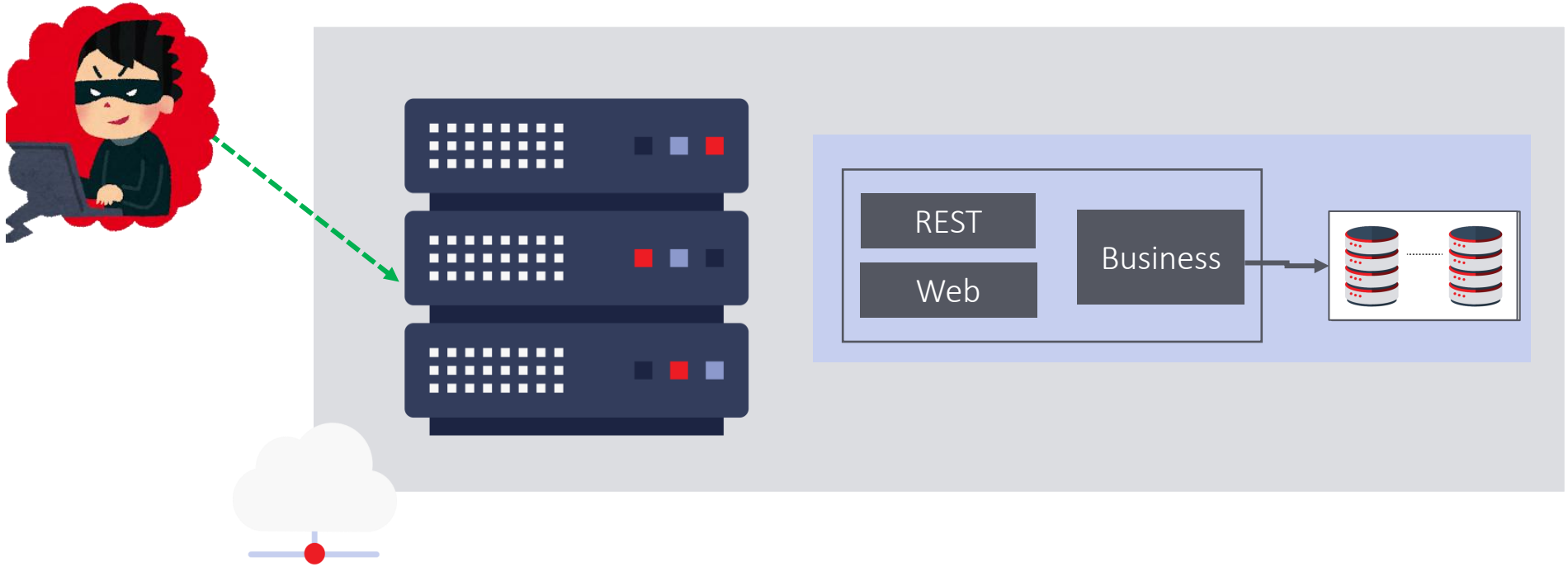


TLS（Transport Layer Security）が以下を保護

- データが暗号化される
- 証明書によるRESTサービスの認証
- 信頼性の高い接続で改ざんを防止

SBI R3 Japan

課題と対策②：物理的なアクセス防御



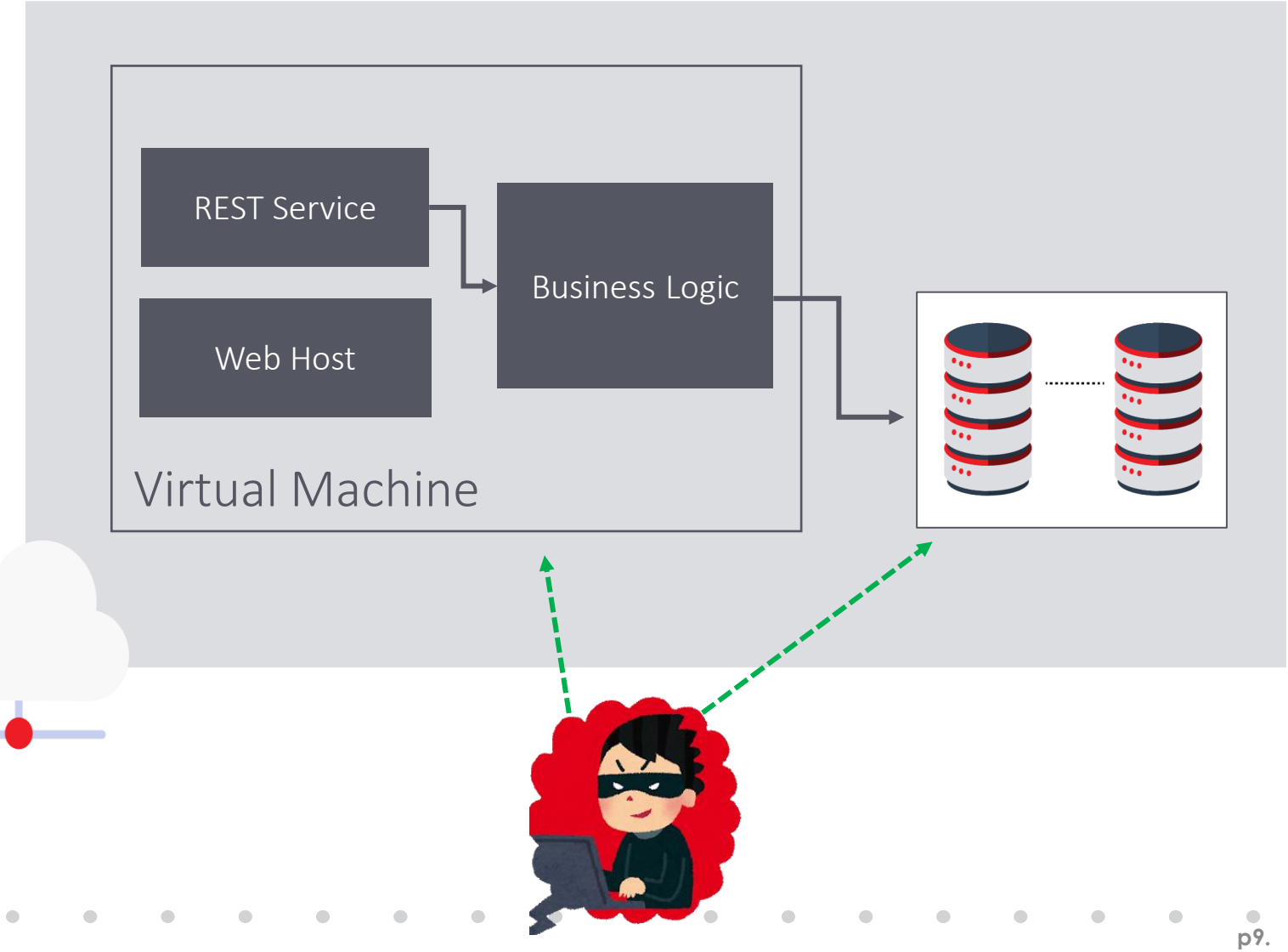
暗号化により、ディスクやファイルストレージへの直接アクセスを防止

- フルディスク暗号化により、ディスクをブロックレベルで保護
- ファイルの内容も暗号化可能
- シャーディングによるデータのサーバーにおける分散化

SBI R3 Japan

課題③：クラウドアクセス

- 入札を記録するコードの修正
- データベースレコードの修正



これを防ぐ手立ては？

SBI R3 Japan

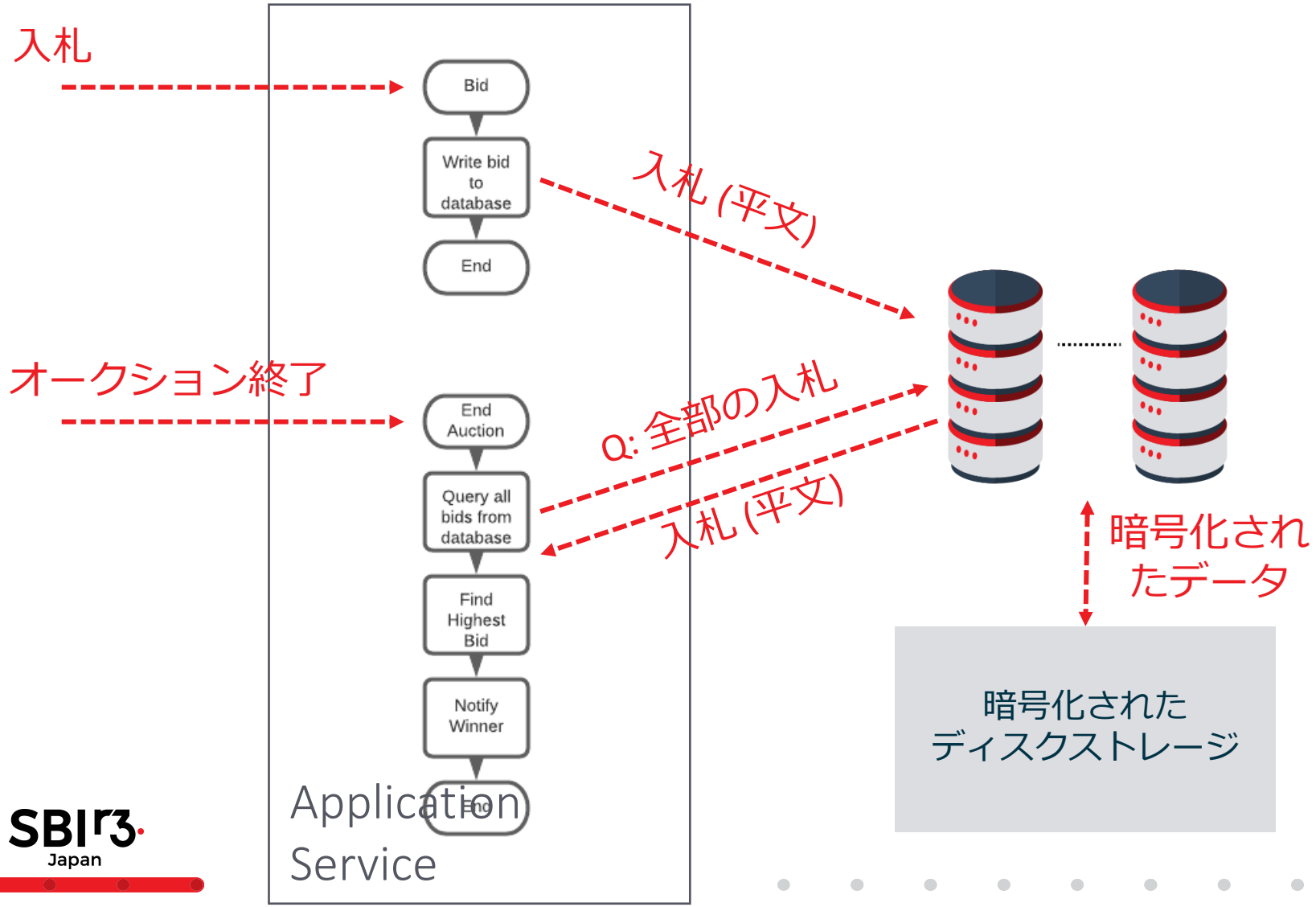
攻撃者は誰？？？

I.　ハッカー：パスワード保護を破り、脆弱性を突く
II.　内部犯行：クラウドサービスの従業員
III. ソフトウェアベンダー：データから最も多くの利益を得る

<u>Ⅲソフトウェアベンダーを信用できない場合</u>
（彼らは一番仕組みに詳しい存在）

SBI R3 Japan

ビジネスロジックは暗号化できない



- 生の入札データにアクセスできなければ、サービスは機能しない
- 使用中のデータ保護

まとめ：データ保護の分類

攻撃	分類	対策方法
ネットワークパケット盗聴/ 中間攻撃者	送信中のデータ	TLS – 暗号化
システムやファイルへの 物理接続	静止データ	フルのディスク/ファイル暗号化 Encrypted shards HSM/BYOK
実行中サービスへの接続	使用中のデータ	ゼロ知識証明 準同型暗号 マルチ・パーティ・コンピューテーション 秘密計算

SBI R3 Japan

SBI r3 Japan

関連ソリューション比較

ZKP/FHE/sMPC/
コンフィデンシャルコンピューティング（C.C.）

ゼロ知識証明（ZKP）

データを公開することなく、データへのアクセス権を持つ事を証明する

例）
- 誕生日を開示せずに未成年でない事を証明する。
- 収入金額を開示せずに住宅ローンの審査基準を満たす収入がある事を証明する。
- ゼロ知識認証（パスワードを送付しない形のログイン）

制限
- **計算能力への高い要求**
- スケーラビリティ
- **実装が難しい（任意のアルゴリズムを実装できるわけではない）**

SBI R3 Japan

準同型暗号化（HE/FHE）

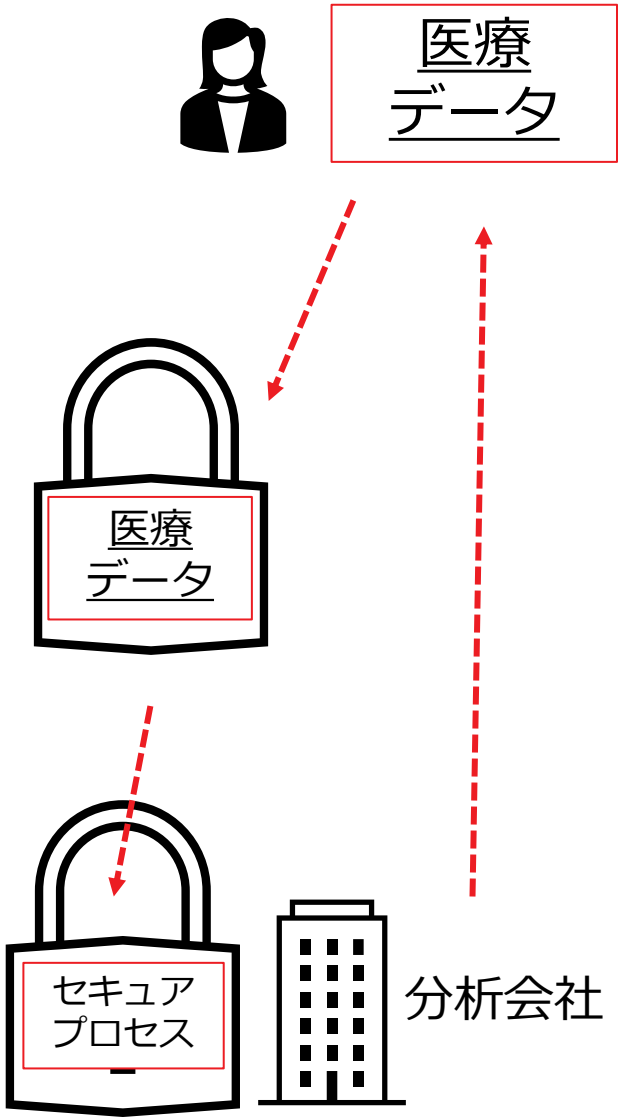
1. 部分準同型暗号(HE)
   - ✓ 単一の数学関数、例：「足し算」
2. 完全準同型暗号(FHE)
   - ✓ 掛け算や足し算 -> NAND ゲート
   - ✓ NANDゲートの組み合わせでフルロジックを構築可能

ユースケース

- ✓ ゼロ知識証明と同じ課題
- ✓ 生データを共有しづらいユースケースでのAI適用
  - ➢ 直近金融系で検討が始まっている…

制限

- 計算コストが高い（クラウド移行すら厳しい）
- 完全準同型暗号は、平文操作の除外が難しく、ユースケースに制限



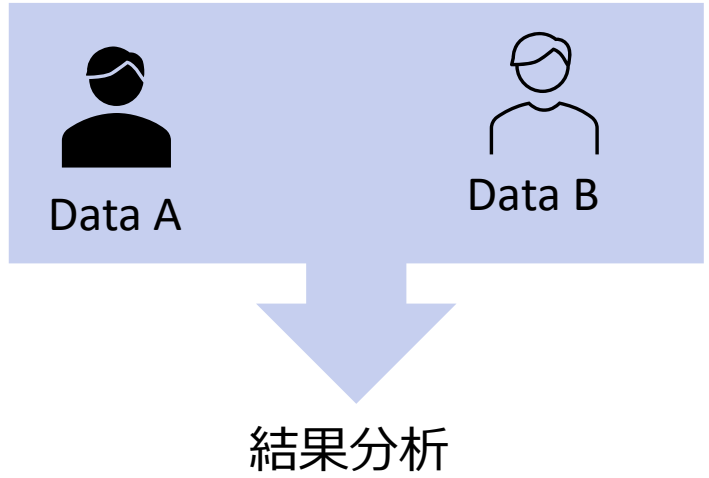
SBI R3 Japan

セキュア・マルチ・パーティ・コンピュテーション (sMPC)

- 各当事者がデータの断片だけを保持する**「秘密の共有」**に依存

ユースケース

- まだこれから探していく
- 個人情報に関わるデータ分析がメインターゲット



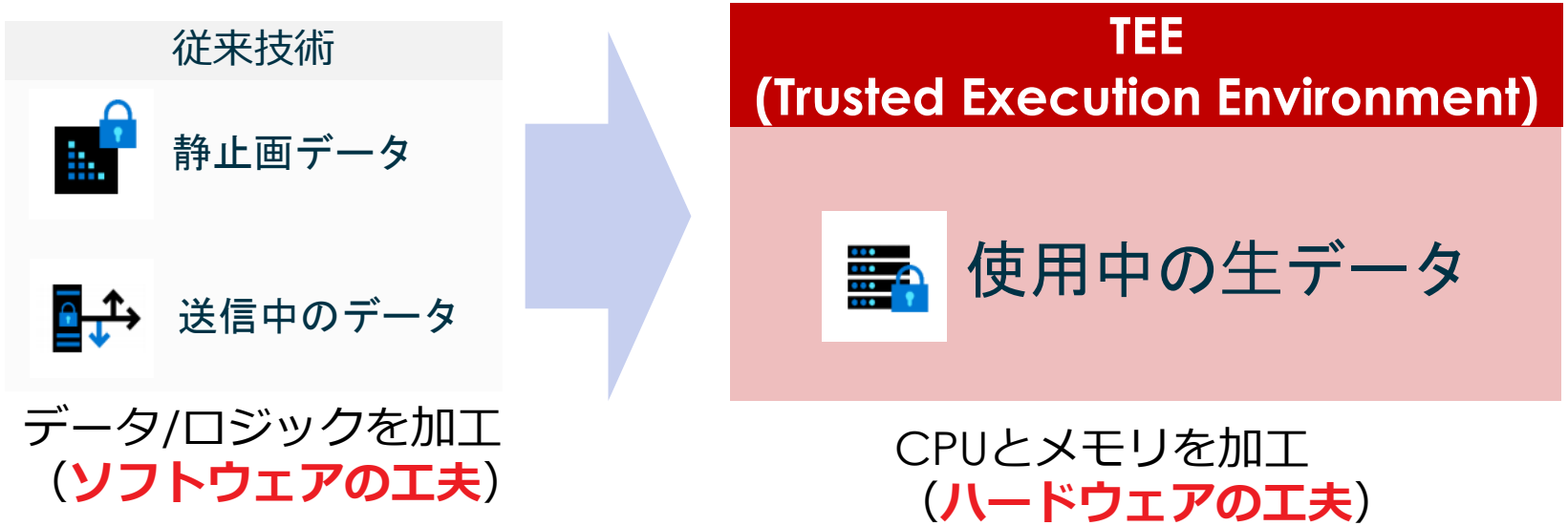
制限

- **ユースケースごとのカスタムアルゴリズム構築**
- 高い通信コスト／高い計算コスト

SBI R3 Japan

コンフィディンシャル・コンピューティング（C.C.）とは？

- **ハードウェアに依拠する全く異なるアプローチ**
- **コード実行＆メモリを他から「分離」する**
  - ✓ データの完全性
  - ✓ データの機密性
  - ✓ コードの完全性



SBI R3 Japan

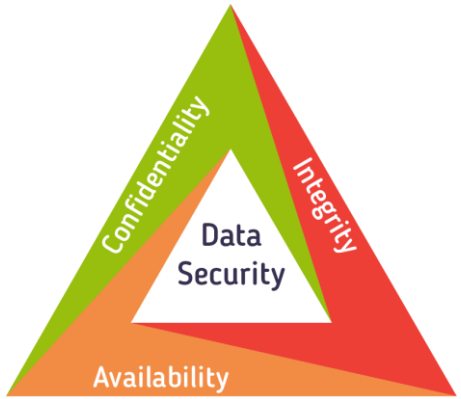
C.C.＝TEEハードウェア が提供する保証

データセキュリティ３要素

✓C：Confidentiality

✓I：Integrity

✓A：Availability



データの完全性（I）	データの不正な変更に対する保護
データの機密性（c）	不正なアクセスからの保護
コードの完全性（I）	実行されるコードの変更防御

SBI R3 Japan

TEE （ハードウェア）ベンダー

TEEベンダーは、以下のように複数あります。

ベンダー/製品	アーキテクチャ	可用性
Intel SGX ※	Intel X86/X64	サーバークラスのCPUを搭載。 • **Azureコンフィデンシャル VM** • ベアメタル/セルフホステッド • Alibaba？
AMD SEV	Intel X86/X64	GCPのリフト＆シフト・ホスティング ベアメタル/セルフホステッド
IBM HyperProtect	z/Architecture	LinuxONE、ホステッドワークロード
ARM TrustZone	ARM	組み込み、IoT
Amazon Nitro Enclaves	Nitro	AWS：すべてのNitro対応VM

※クライアント／ホスト／Enclaveの完全なソリューションに必要な多くのセキュリティ機能を提供しているのは、今のところIntel SGXのみ。

SBI R3 Japan

SBI r3 Japan

Conclave/Enclaveのご紹介

- Conclave特徴紹介
-
-
-

Conclaveとは？

ここまでの振り返り

- 解きたい課題：例：封印入札オークション
- 課題解決方法
  - ✓ZKP　✓HF/FHE　✓sMPC
  - ☑コンフィデンシャル・コンピューティング（C.C.）
- C.Cを実現するハードウェア＝T.E.E
  - ✓Intel SGX（製品名称）
  - ✓**Enclave**
  - ✓**Conclave**

SBI R3 Japan

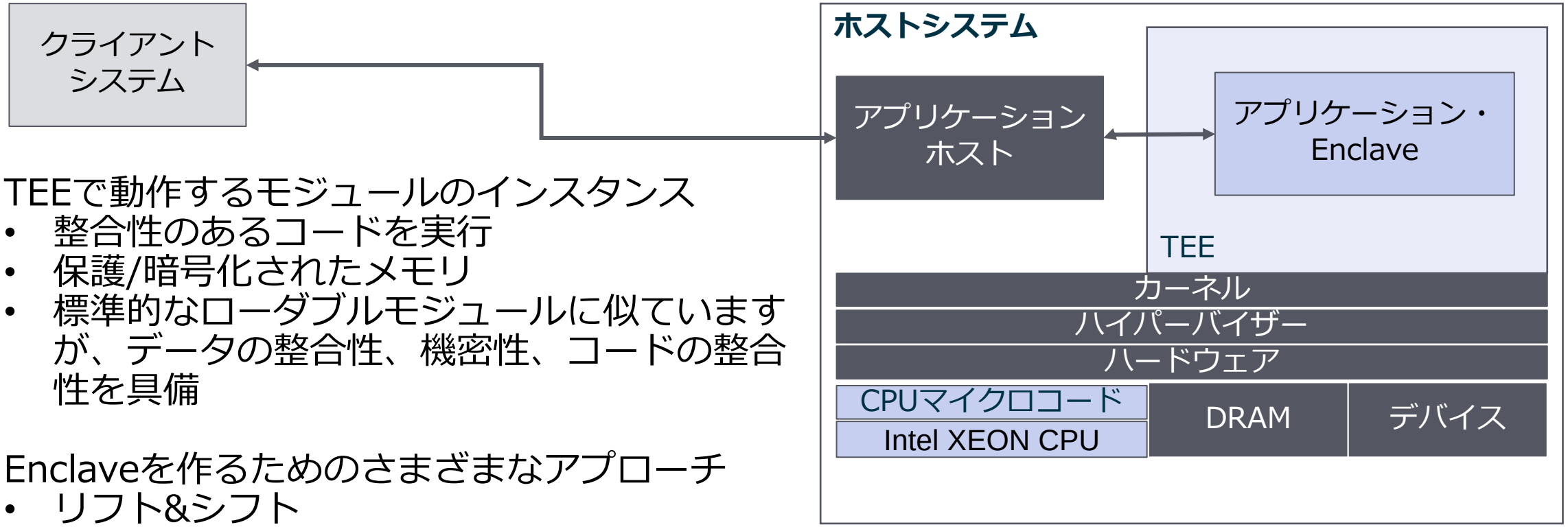
Conclaveとは？

ここまでの振り返り

- 解きたい課題：例：封印入札オークション
- 課題解決方法
- C.Cを実現するハードウェア＝T.E.E
  - ✓Intel SGX（製品名称）
  - ✓**Enclave　←　実際に生データが展開される領域**
  - ✓**Conclave ←　上記のTEE実装を容易にするSDK**

SBI R3 Japan

Enclave:イントロダクション



TEEで動作するモジュールのインスタンス
- 整合性のあるコードを実行
- 保護/暗号化されたメモリ
- 標準的なローダブルモジュールに似ていますが、データの整合性、機密性、コードの整合性を具備

Enclaveを作るためのさまざまなアプローチ
- リフト&シフト
- プライバシー関連コードのカプセル化

SBI R3 Japan

Enclaveの実装法①リフト&シフト

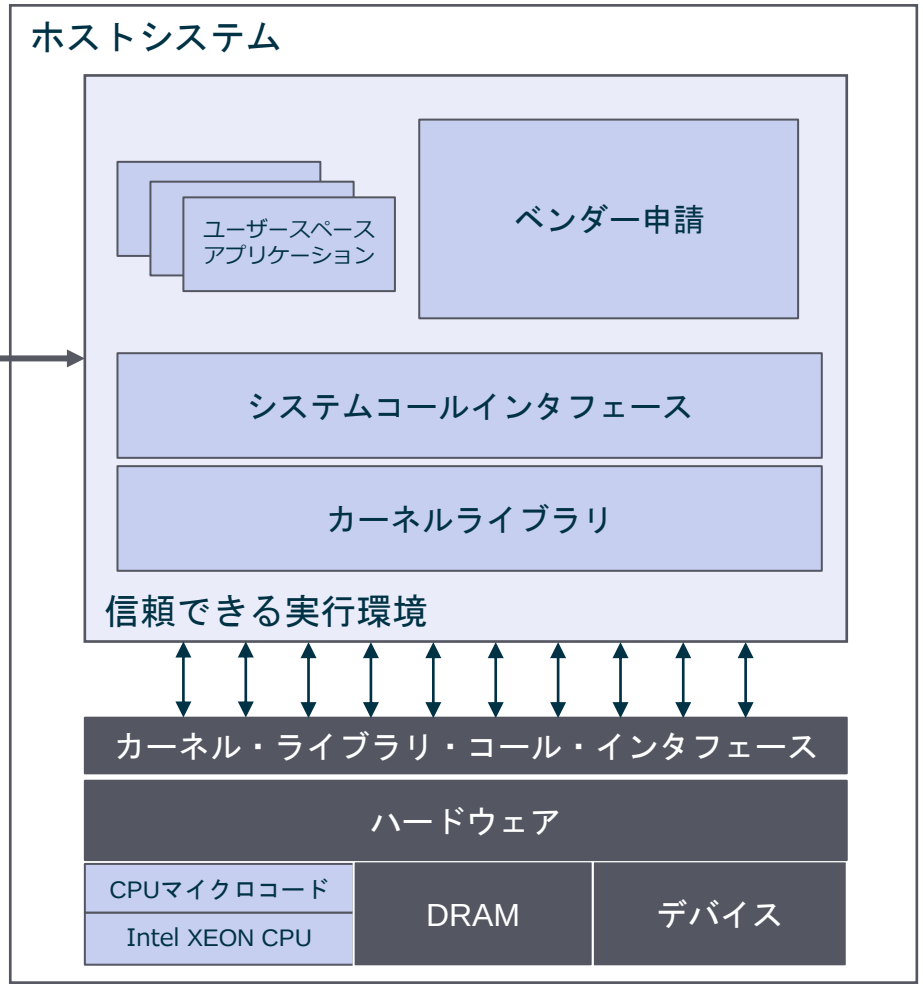
クライアント
システム

○メリット

- アプリケーション移管が容易

×デメリット

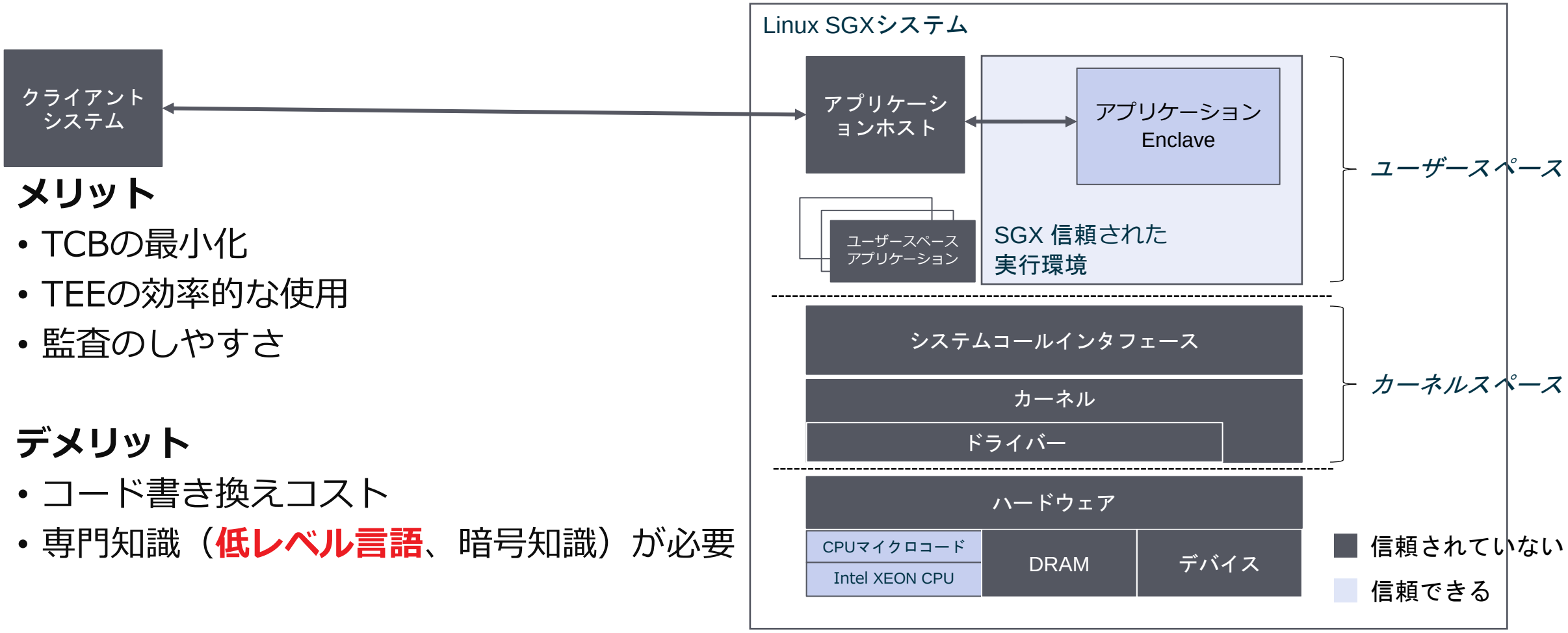
- 複雑で多様な攻撃対象
  - ✓ TCB、アタッキングフェース
- 信頼できる管理者の確保が必要
- 監査が困難



信頼されていない

信頼できる

Enclaveの実装法②:カプセル化



メリット

- TCBの最小化
- TEEの効率的な使用
- 監査のしやすさ

デメリット

- コード書き換えコスト
- 専門知識（**低レベル言語**、暗号知識）が必要

SBI R3 Japan

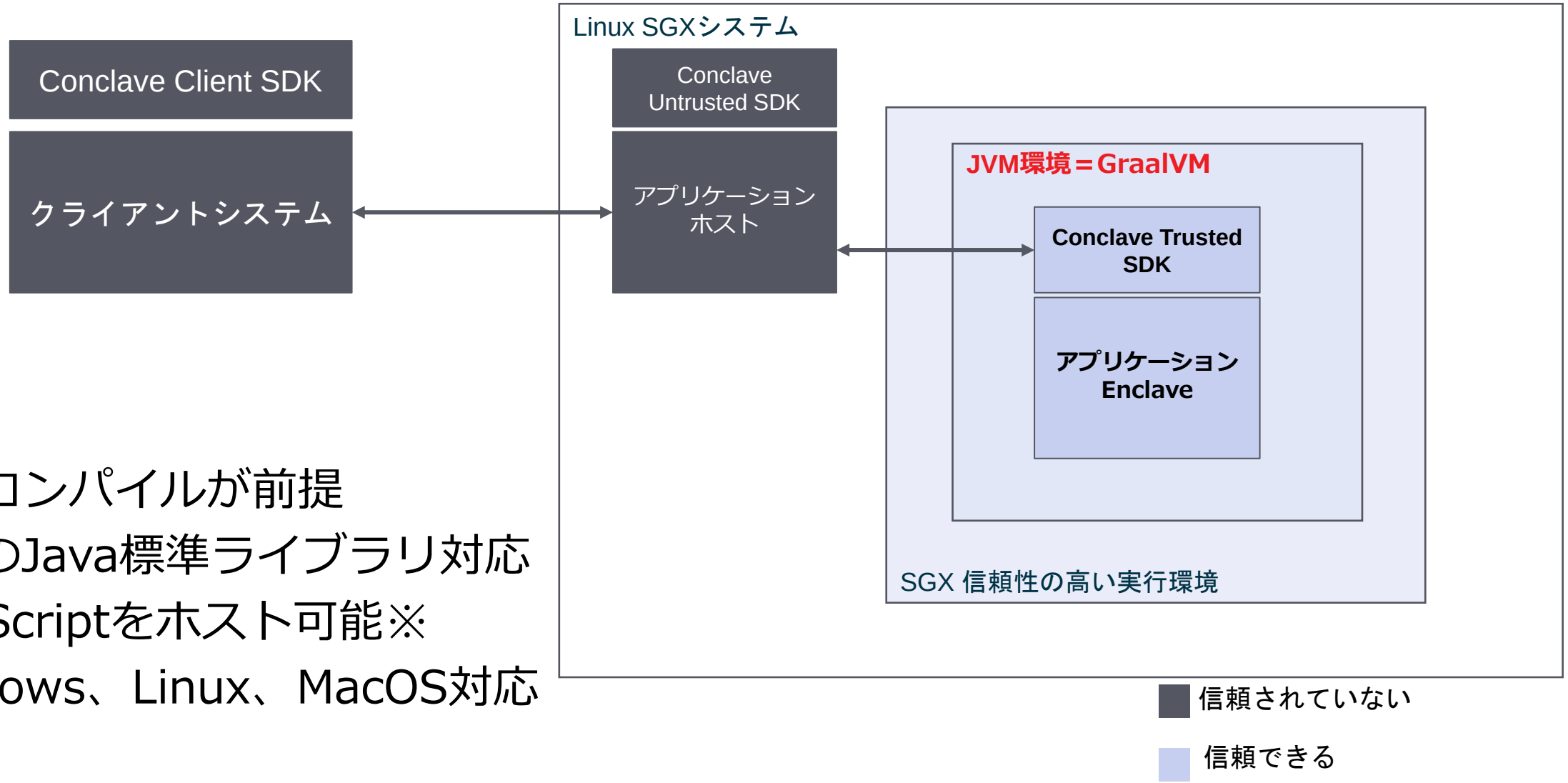
SBI r3 Japan

conclave

Conclaveは、インテルSGXを使ったアプリケーション開発を容易にします。

- JavaやKotlinなどの高レベル言語でアプリケーションを構築
- 使いやすくなった高レベルAPI
- SGXのベスト・プラクティスに沿ったAPIデザイン
- インターネットを介したEnclaveの監査を完全にサポート

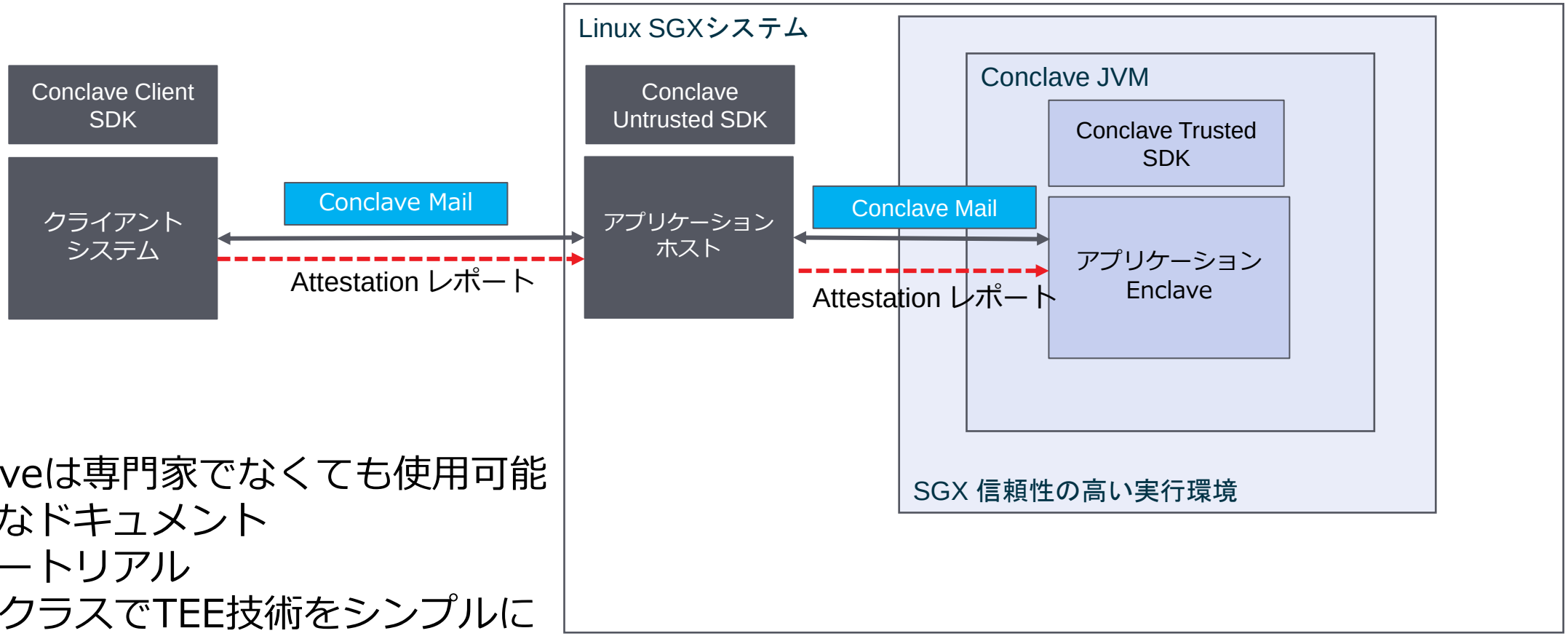
Conclaveの特徴 :Graal VM Native Image



- AOTコンパイルが前提
- 多くのJava標準ライブラリ対応
- JavaScriptをホスト可能※
- Windows、Linux、MacOS対応

※動的言語の利用はセキュリティ上注意が必要

Conclaveの特徴



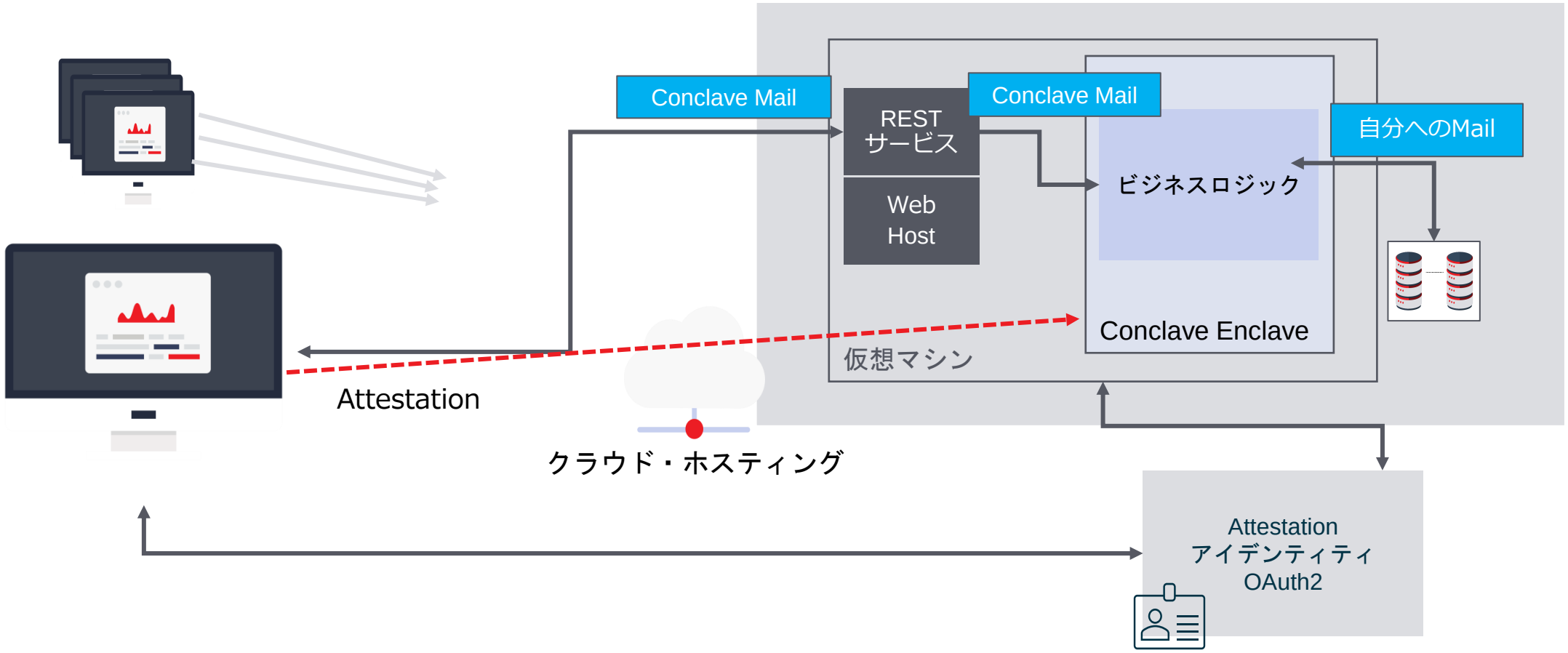
Conclaveは専門家でなくても使用可能

- 明確なドキュメント
- チュートリアル
- SDKクラスでTEE技術をシンプルに
- 簡単なデプロイメント

信頼されていない
信頼できる
暗号化

SBI R3 Japan

Conclave事例：封印入札アプリケーション



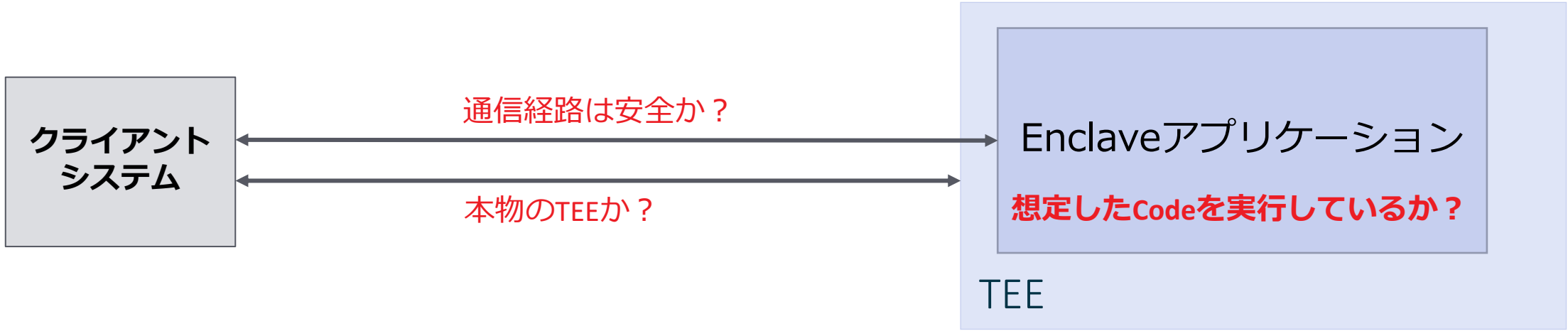
SBI R3 Japan

SBI r3 Japan

Conclave/Enclaveのご紹介

- Conclave特徴紹介
- Attestationとは？
-
-

Enclaves: Clientシステムから見た信頼を確保するには？



（Attestation、４つの論点）

① TEEのハードウェアが本物であることをクライアントに証明する。

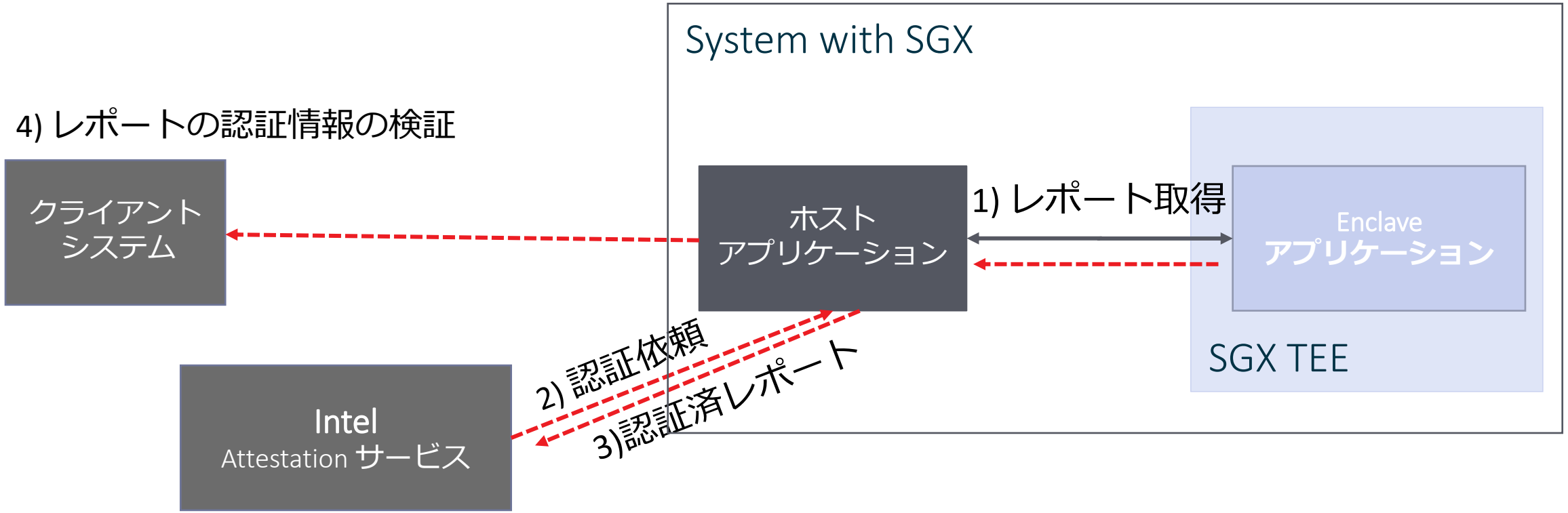
② Enclaveが正しいコードを実行した事を証明する

③ Enclaveだけが送信データにアクセス可能だと確認する。⇒ Conclave Mail

④ クライアント・アプリケーションが危険にさらされていない事を確認する。

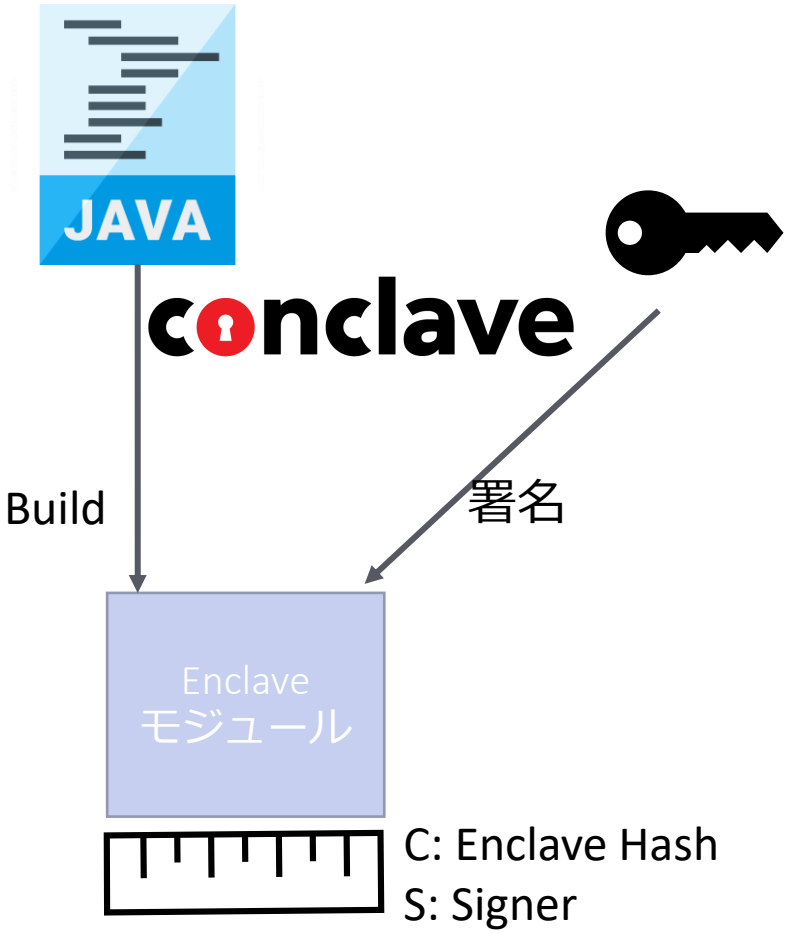
SBI R3 Japan

Attestation① TEE（ハードウェア）を使用していることの証明



- ✓**Intel社がSGXの信頼の根源になる。**
- ✓ **Conclaveは、EPID/DCAPに対応。**

Attestation② Enclaveが正しいコードを実行した事を証明する



Conclaveアプリ構築プロセス

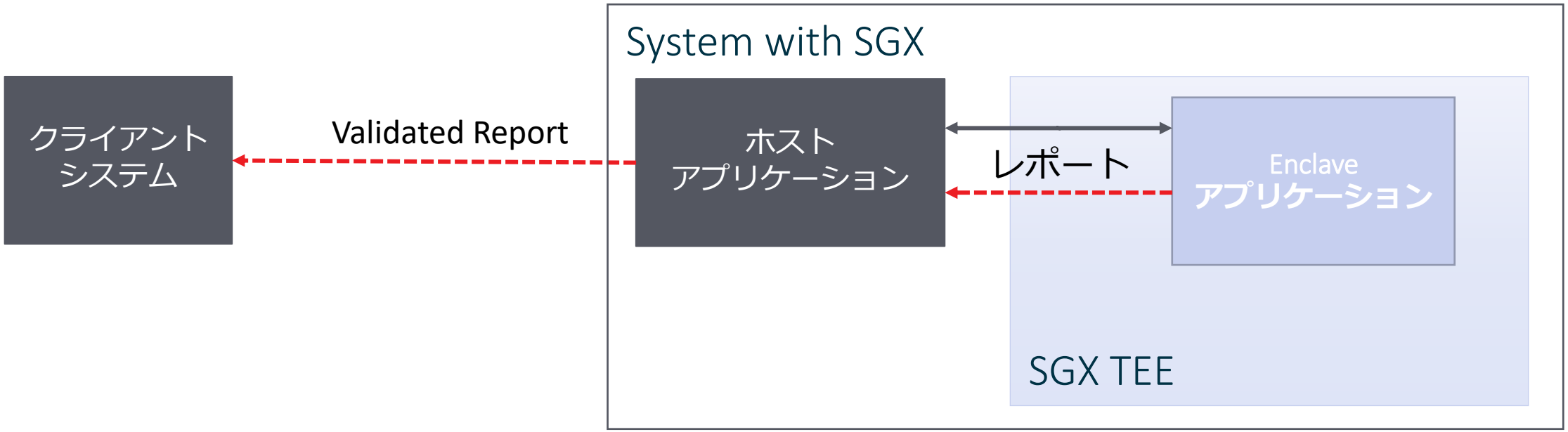
１. Java/Kotlinのソースがコンパイルされ、Enclaveモジュールが作成される

２ Enclaveモジュールに署名する（Buildを行った主体の秘密鍵で）

⇒検証に使用する
２つのデータ＝「制約」が得られる
☑Enclaveモジュールのハッシュ
☑署名者の公開鍵

SBI R3 Japan

Attestation② Enclaveが正しいコードを実行した事を証明する



Conclaveが「制約」に関する簡単なDSLを提供

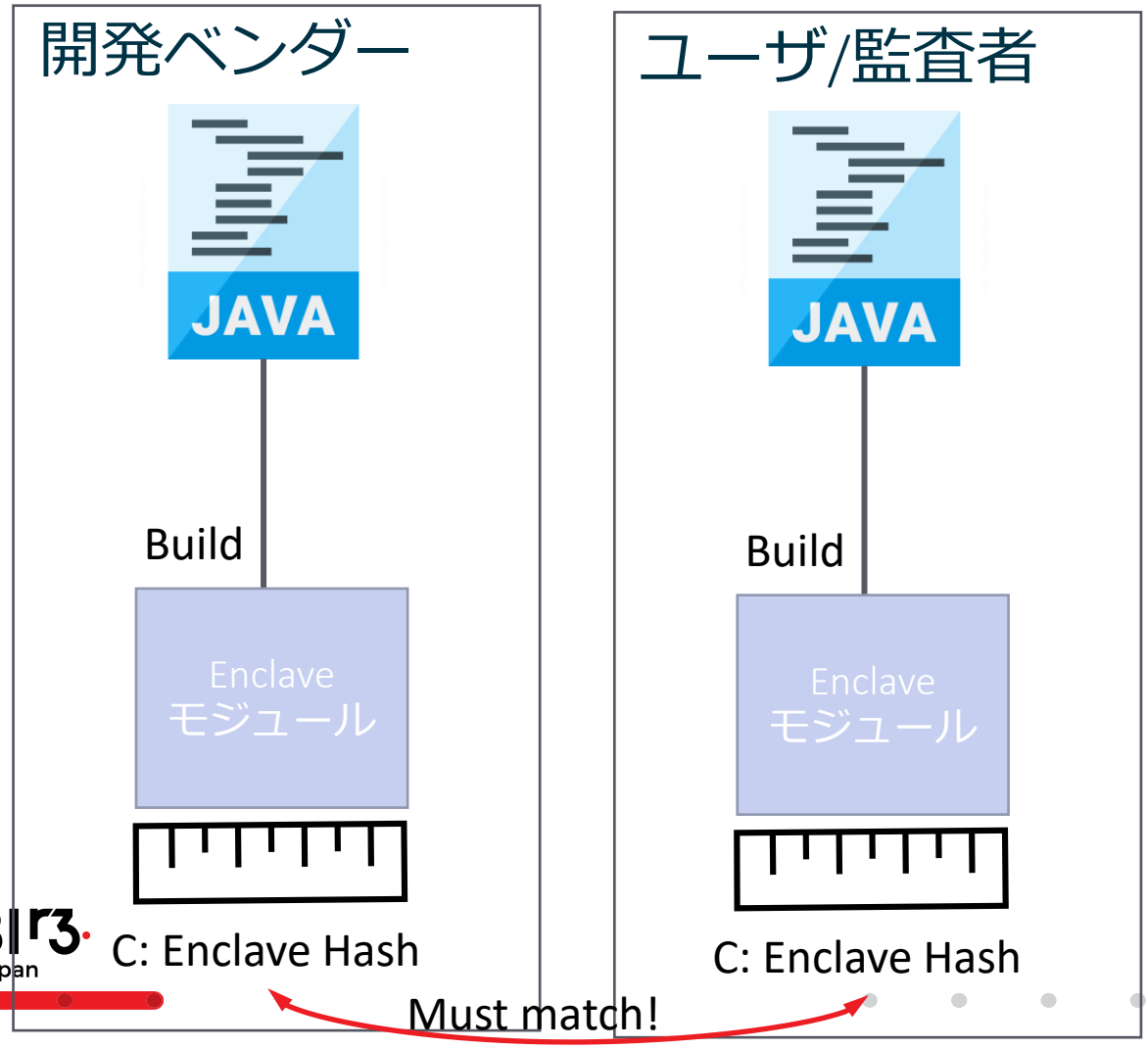
(DSL定義)

S: Enclaveに署名した公開鍵のハッシュ値　C: Enclaveモジュールのハッシュ値

PROD: Product ID　　SEC: SGX のセキュリティ設定値（プロダクション／ノンプロダクション

SBI r3 Japan

Attestation② Enclaveが正しいコードを実行した事を証明する
～コード監査と複数回構築（Repeatable Build）　概念レベル＆未実装～



Enc.モジュールの確認性確認Step

1. 開発ベンダーがバイナリイメージをビルド
2. ユーザ/監査者がソースコードを監査
3. ユーザ/監査者が手元でビルドして、ハッシュの一致を確認
4. ユーザ/監査者が署名を作成
   ※ハッシュ値とセットで「制約」になる

5. 「制約」をクライアントアプリケーションに登録

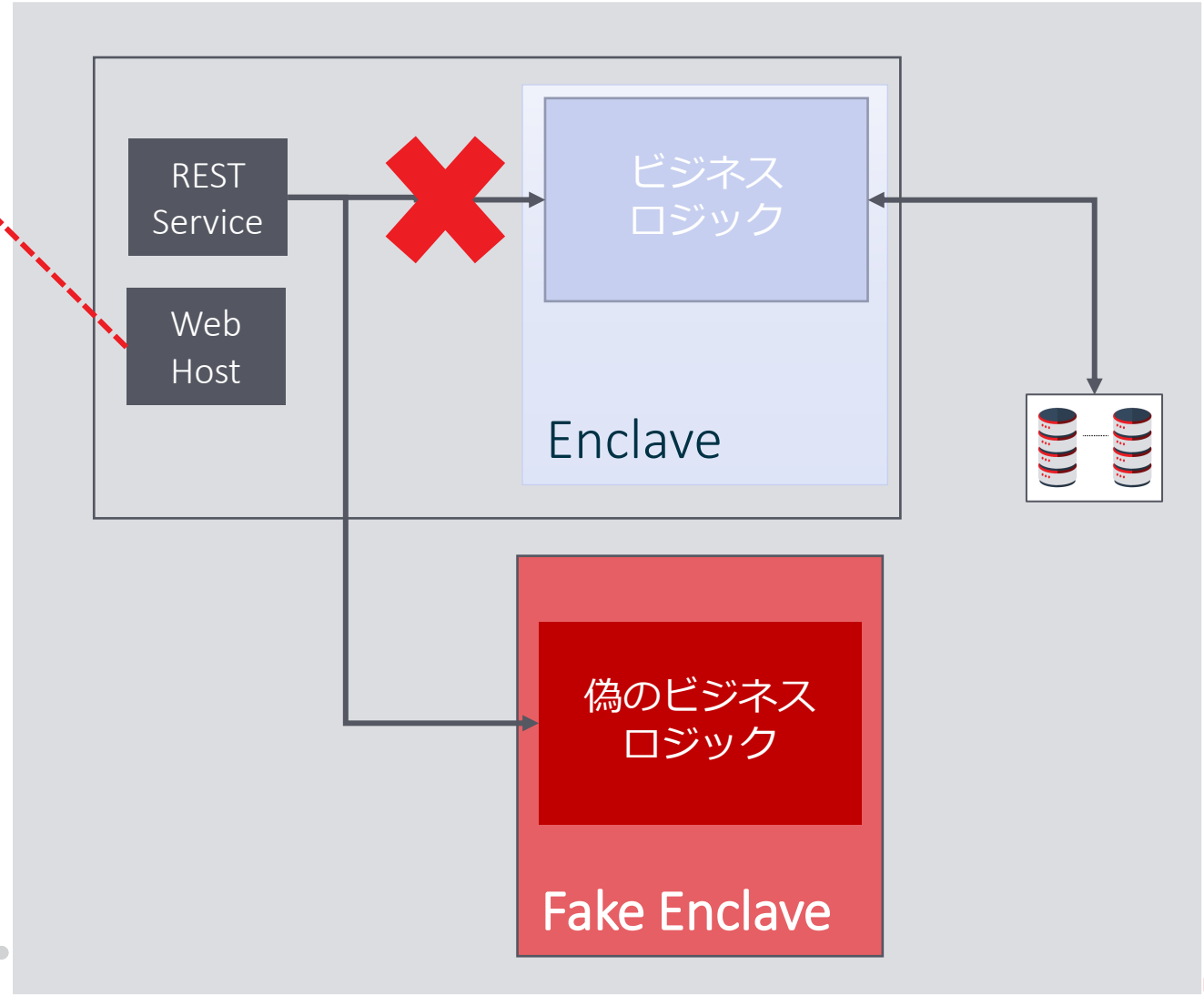
Attestation② Enclaveが正しいコードを実行した事を証明する
～コード監査と複数回構築（Repeatable Build）～

Repeatable Buildの要否は、信頼関係次第

クライアントが信頼する相手	モジュールBuild	モジュール署名	repeatable build要否
ベンダーと監査者が共謀しないと信じる	ベンダー、監査者	監査者	Yes
ベンダー	ベンダー	ベンダー	No
監査者	監査者	監査者	No
クライアント	クライアント	クライアント	No

SBI R3 Japan

Attestation④ クライアント・アプリケーションの課題



クライアントアプリが「偽物」のリスク

⇒「偽」Conclaveへ接続するリスク

対策：
WebからDLするのではなく
署名付きInstallableアプリにする。

SBI R3 Japan

SBI R3 Japan

Conclave/Enclaveのご紹介

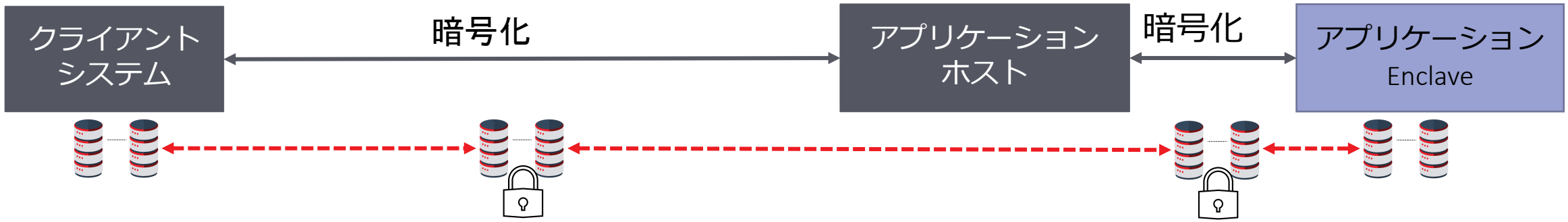
- Conclave特徴紹介
- Attestationとは？
- Conclave Mail

Enclaveとのデータ交換

クライアント／Enclave間の通信は暗号化が必要

なぜTLSを使わないのか？

- ✓ 詳しくは以下参照：https://docs.conclave.net/mail.html#tls-complexity
- ✓ TLSはEnclaveに大きなTCBを必要とする

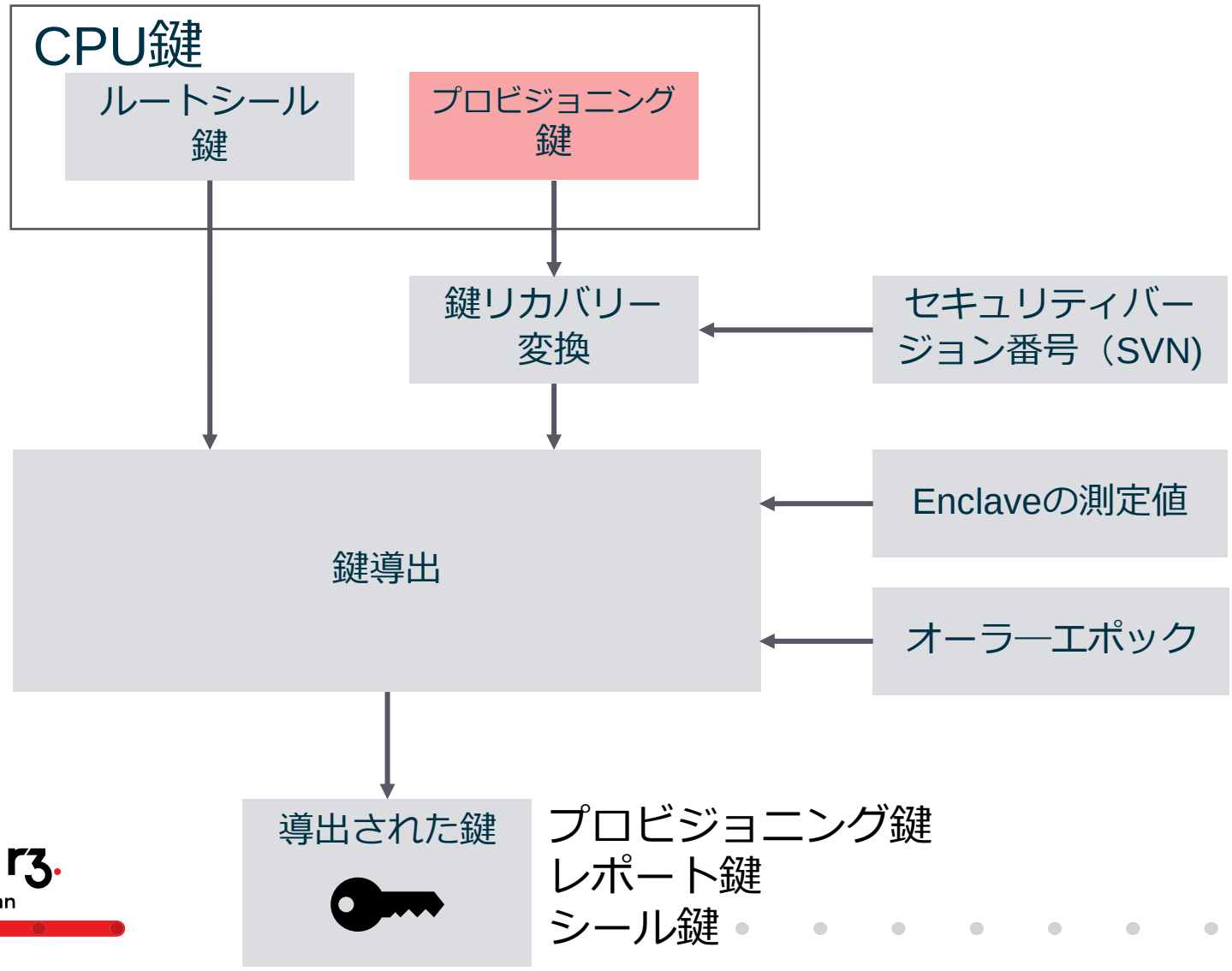


Conclave Mail

- Noiseフレームワーク: WhatsAppで長年使用され信頼の高い暗号化通信
- 信頼はEnclaveの鍵に根ざしている
- 小さなTCB

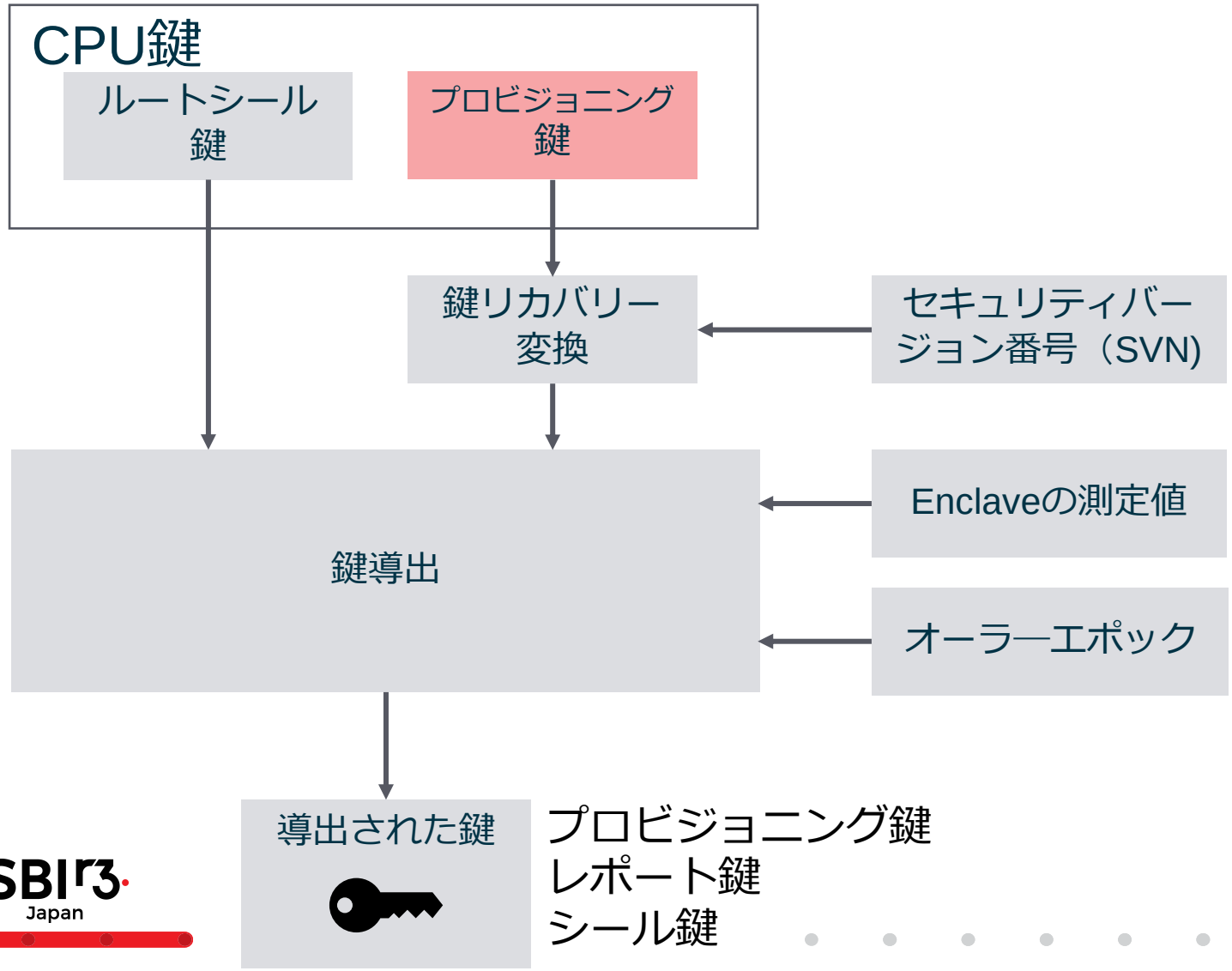
SBI R3 Japan

Enclaveでのデータ交換：鍵１



- CPUには製造時にキーが付与
- ルートシール鍵はCPUのみが知る
- プロビジョニング鍵はCPUとIntelが知りえ、信頼の根源(RoT) となる
- SVNとプロビジョニング鍵を組み合わせてセキュリティリカバリーに使用

Enclaveでのデータ交換：鍵2



Enclave・キーは複数の入力から生成

- プロビジョン鍵：プラットフォームのプロビジョニング
- レポート鍵：Attestation
- シール鍵：Enclave外のデータを保護する

コンクレーブは、以下からのシール鍵を使用

- ルートシール鍵
- 現在のSVNを含むプロビジョニング鍵
- Enclaveの署名鍵
- Enclaveのバージョン (ISVSVN)

Enclave内のデータは以下でのみアクセス可能

- 同じ署名鍵で署名されたEnclave
- SVNが現在のバージョン以上
- Enclaveのバージョンが現在のバージョン以上

SBI R3 Japan

Enclaveでのデータ交換：なぜ署名鍵なのか

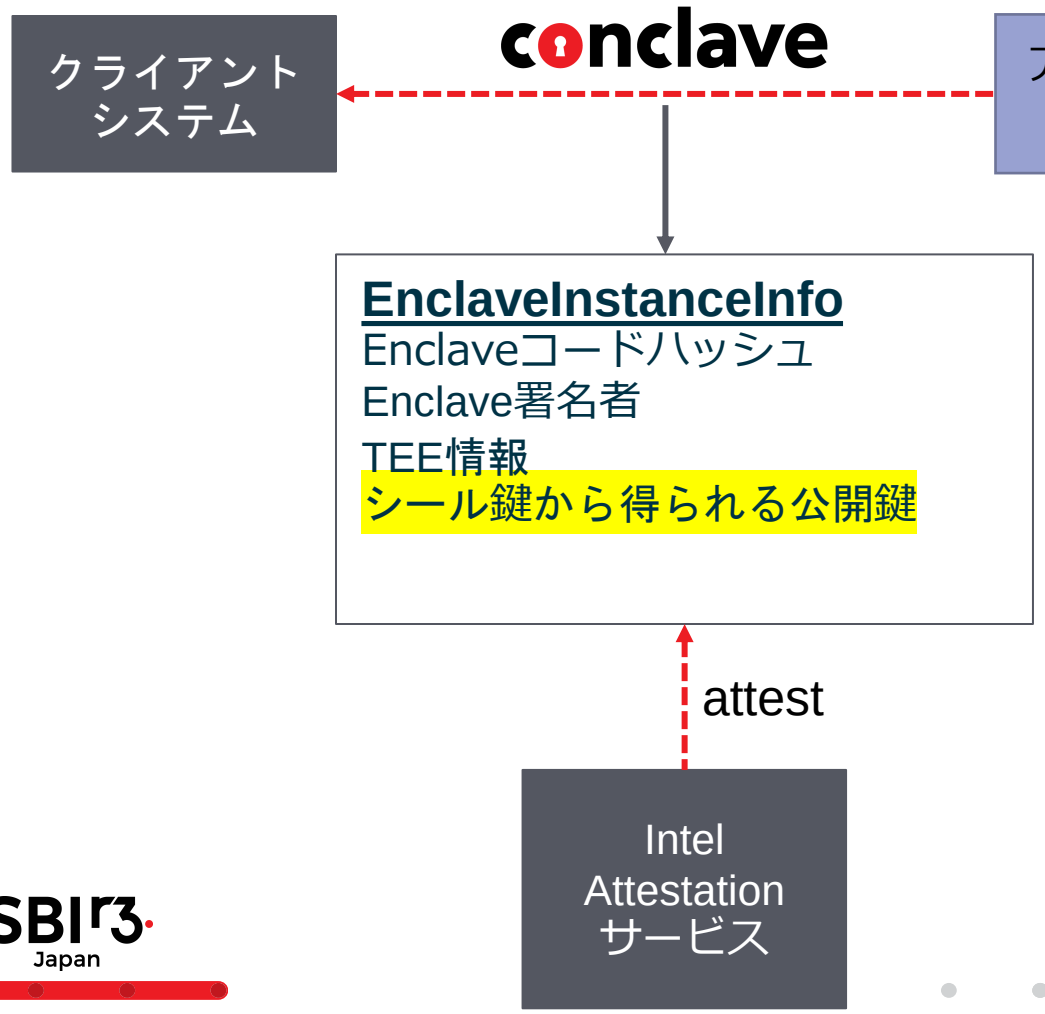
ConclaveはEnclaveの測定値ではなくEnclaveの署名鍵でデータを封印する理由

- バグを修正したり機能を追加するとデータを失う
- Enclave測定値ではEnclaveのバージョン・アップグレードが不可

Enclaveは、お客様の信頼を得られる手順で署名される必要がある

SBI R3 Japan

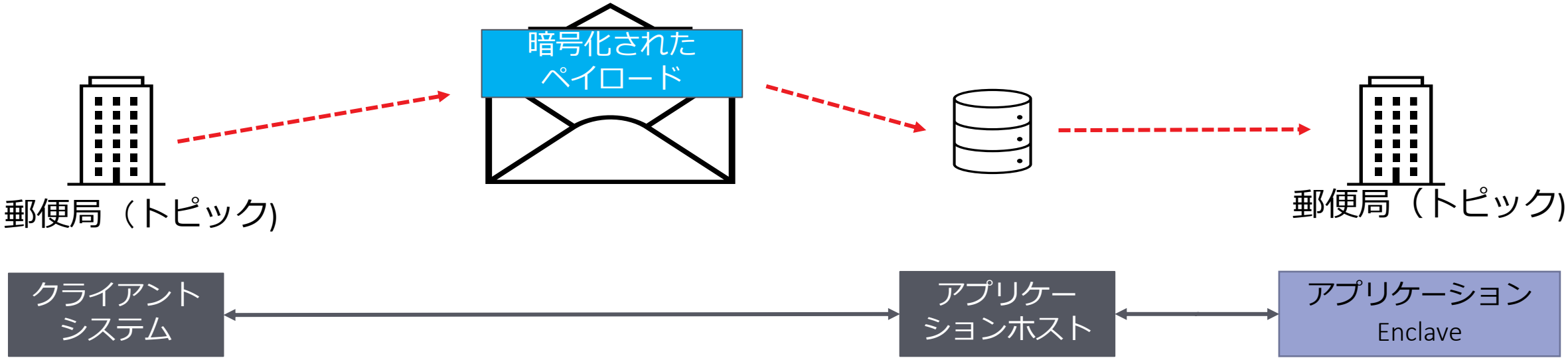
Enclaveでのデータ交換：鍵の交換



EnclaveInstanceInfoは公開鍵を持ち

- シール鍵より導出される
- attestationによる鍵の完全性の検証
- Enclaveのみがその鍵で暗号化されたデータを復号できる

Enclaveでのデータ交換：Conclave Mail

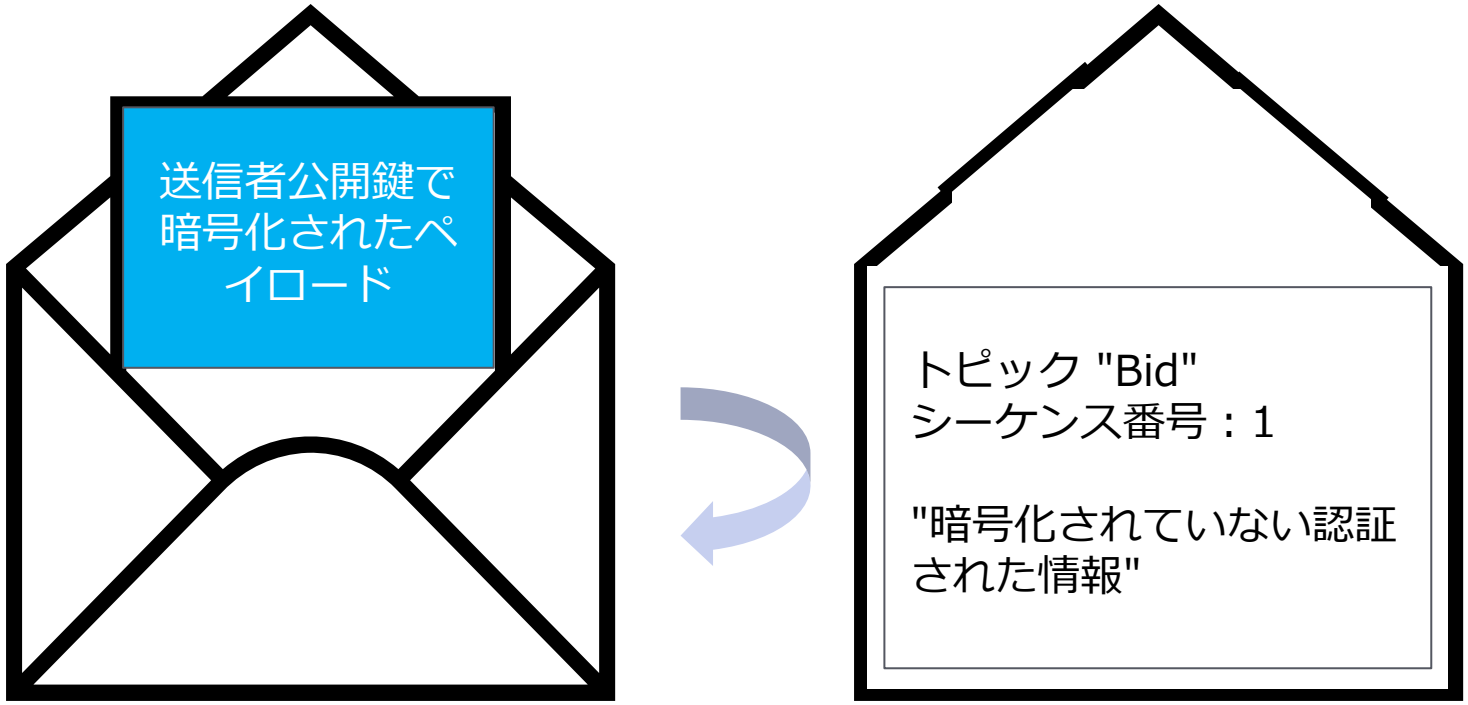


- Conclave Mailのメッセージは、郵便局を使って準備されます。
- 各郵便局は、メールストリームを識別するテキスト「トピック」を持つ
- トピックは送信者ごとに設定されます
- 各トピックは独自のシーケンス番号を持つ

郵便局？
下にある通り、SD
Conclave Post O
ですけど、
ポストオフィスと
す？

SBI R3 Japan

Enclaveでのデータ交換：Conclave Mail



メールの暗号化部分には、受信者の公開鍵で暗号化された以下のものが含まれています。

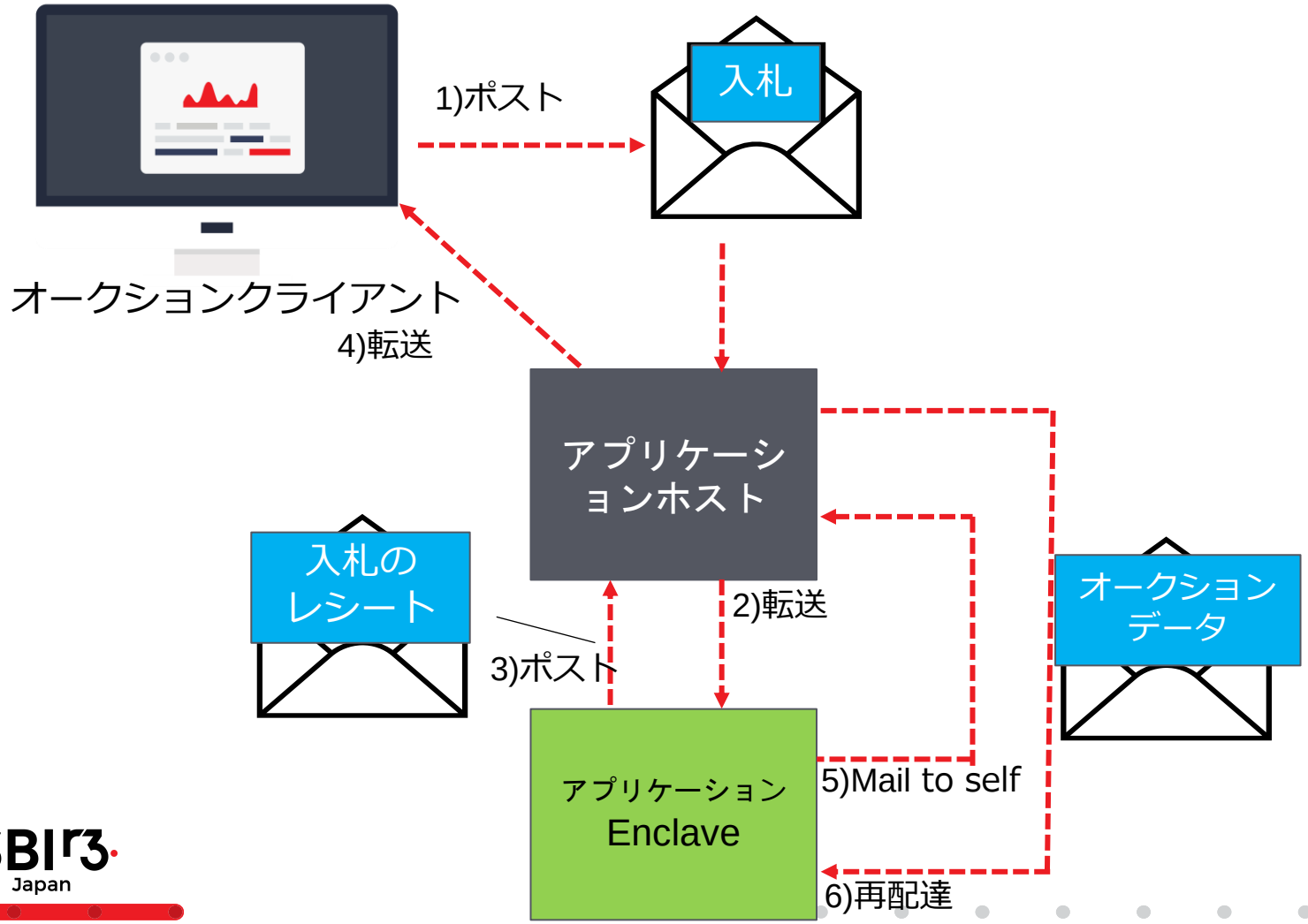
- 送信者の公開鍵
- 暗号化されたペイロード

メールヘッダーには、暗号化されていないが認証された情報が含まれています。

- トピック - ストリーム識別子
- シーケンス番号
- "エンベロープ」情報

SBI R3 Japan

Conclave Mailの利用



Mailの用途は、Enclaveとの間でメッセージを送信すること

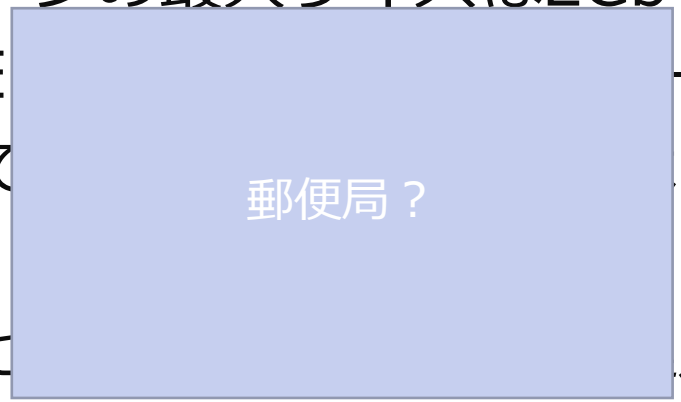
Enclaveは[redacted]ることも可能

郵便局？

- Enclave[redacted]させるた[redacted]
- Enclaveのメモリを解放するために使用できます。
- Mail to selfは、特定のシステム上の単一のEnclaveにバインドされる。

Conclave Mail:考慮すべきポイント

- 異なる物理システム上の同じEnclaveには、異なる鍵が設定されます。
- 現在のところ、Mailメッセージの最大サイズは2Gbです。
- Mailのヘッダは、ホストがE ージをキューイングまたはキャッシュするために使用て すぐに配信される必要はありません。
- 悪意のあるホストは、メッセ 延させたり、再生したりすることができます。



SBI R3 Japan

SBI R3 Japan

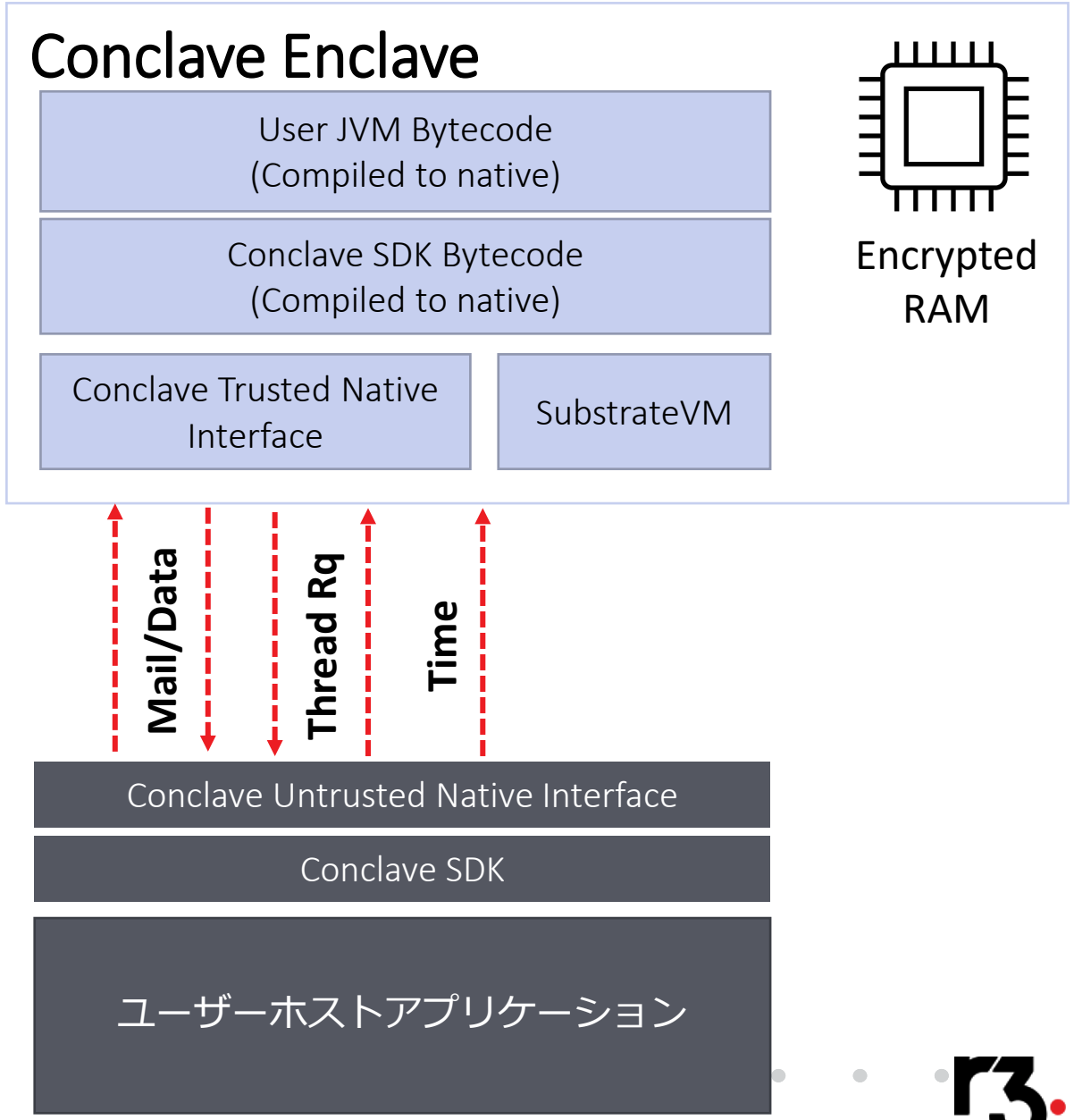
Conclaveアプリケーション構築

設計上の留意点

Conclaveを使う場合の主な制限

- プレコンパイル
  ✓JNIはサポートされない。
  ✓GraalVMの限定的サポート
- 制限されたインターフェース
  ○データ送受信
  ○スレッド化
  ×ハードウェアアクセス
  ×network IFへのアクセス
  ×ファイルシステムへのアクセス
  ×信用できる時間の取得

- 大量のメモリを使うには最新のXeonが必要。



制限事例①：機械学習アプリ

Enclaveで作ったアプリ⇒Conclaveに移す場合・・・・

- ネイティブライブラリの使用（将来バージョンでサポートされる可能性あり）
- ソケット/ネットワークへのアクセス
- ホストファイルシステムへのアクセス

SBI R3 Japan

制限事例②：大規模データ処理

年間5000 件のデータがEnclaveに送付される
×
１件当たり5MB (うち10Kが検索用文字データ)

Enclave
アプリケーション

- データベース全体のサイズ = 5000 * 5MB = ~24GB
- 検索対象 = 5000 * 10K = ~49MB

SBI R3 Japan

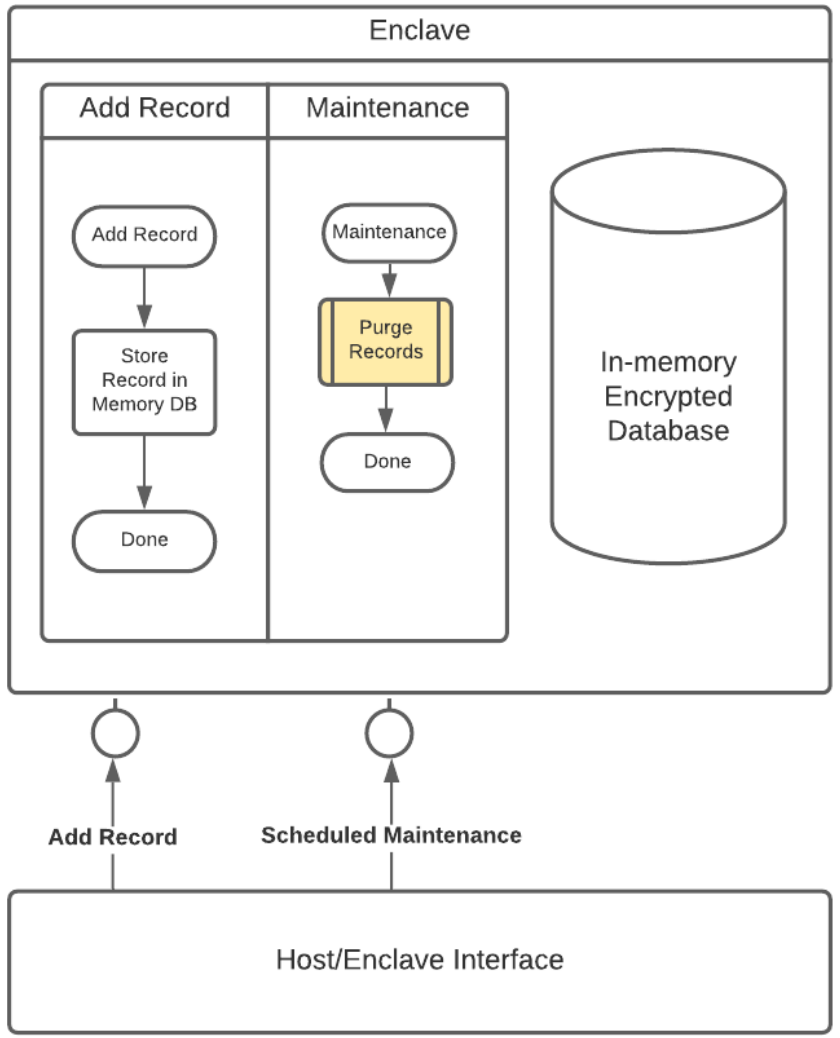
制限事例②：大規模データ処理　：100% オンメモリ

オンメモリーにできる？

1. 最新のXeonなら対応可能
2. ページングを使う
   1. パフォーマンス低下が発生
   2. スケーラブルではない。

5000件/year

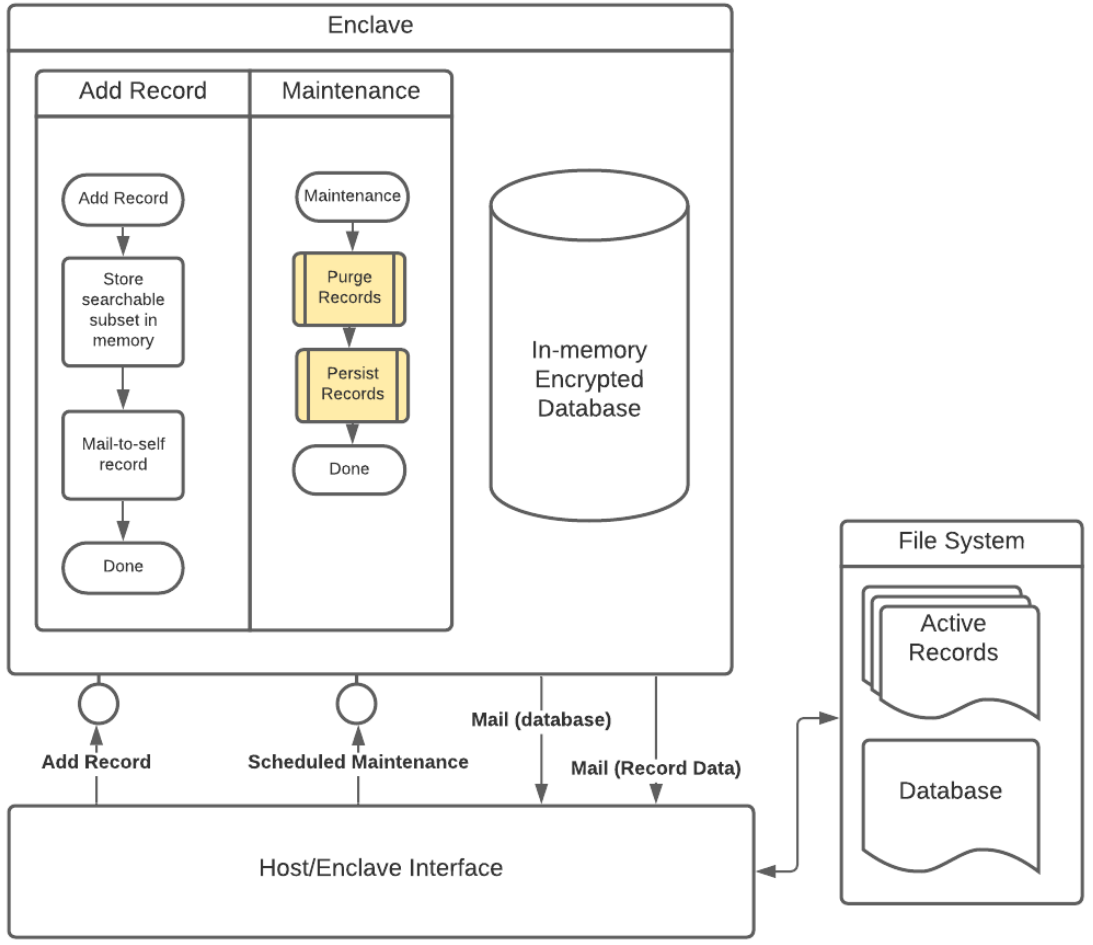
How many queries?



制限事例②：大規模データ処理　：外部DBへ保存

検索可能なデータと静的なデータを分離

- レコード追加時に検索対象データを抽出
- 検索データをインメモリDBに追加
- レコードは“mail-to-self”で外部DBへ
  - レコードは暗号化された状態
- 必要に応じて外部DBからEnclaveへ返送

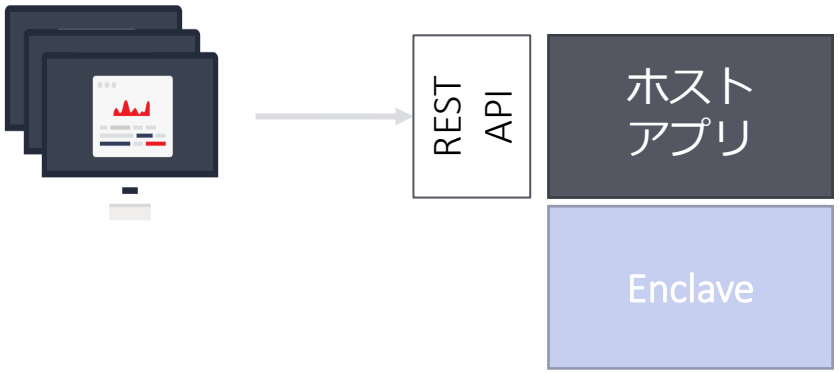


5000件/年

How many queries?

制限事例③：大量処理：外部APIでの工夫

ソリューション①: REST

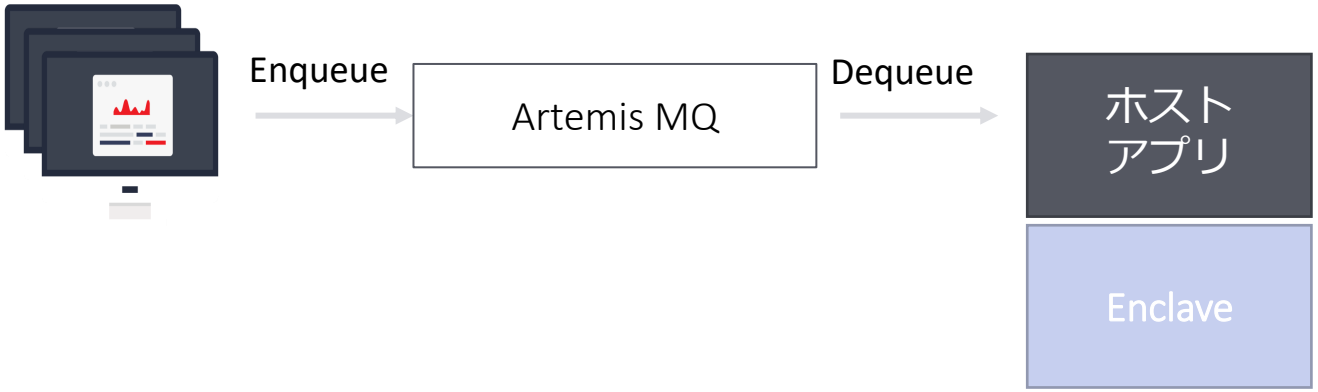


通信量が少ない

✓リソースが予想可能な場合

- ✓CPU
- ✓通信

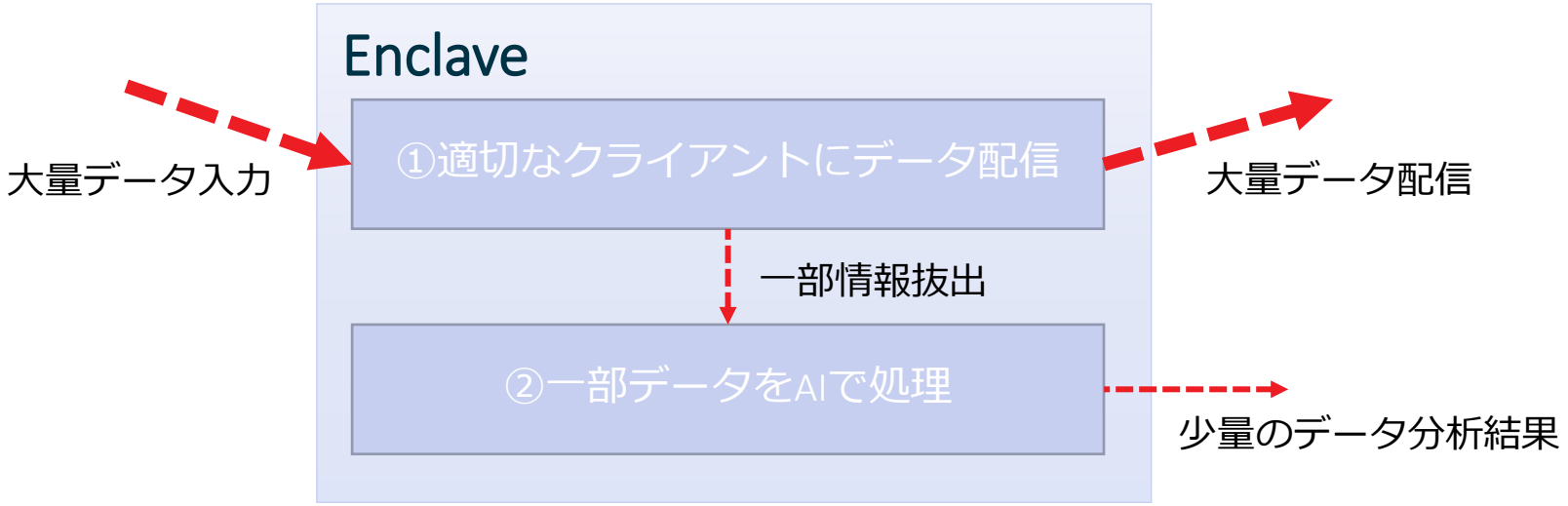
ソリューション②: MQ



負荷が急増する可能性がある場合

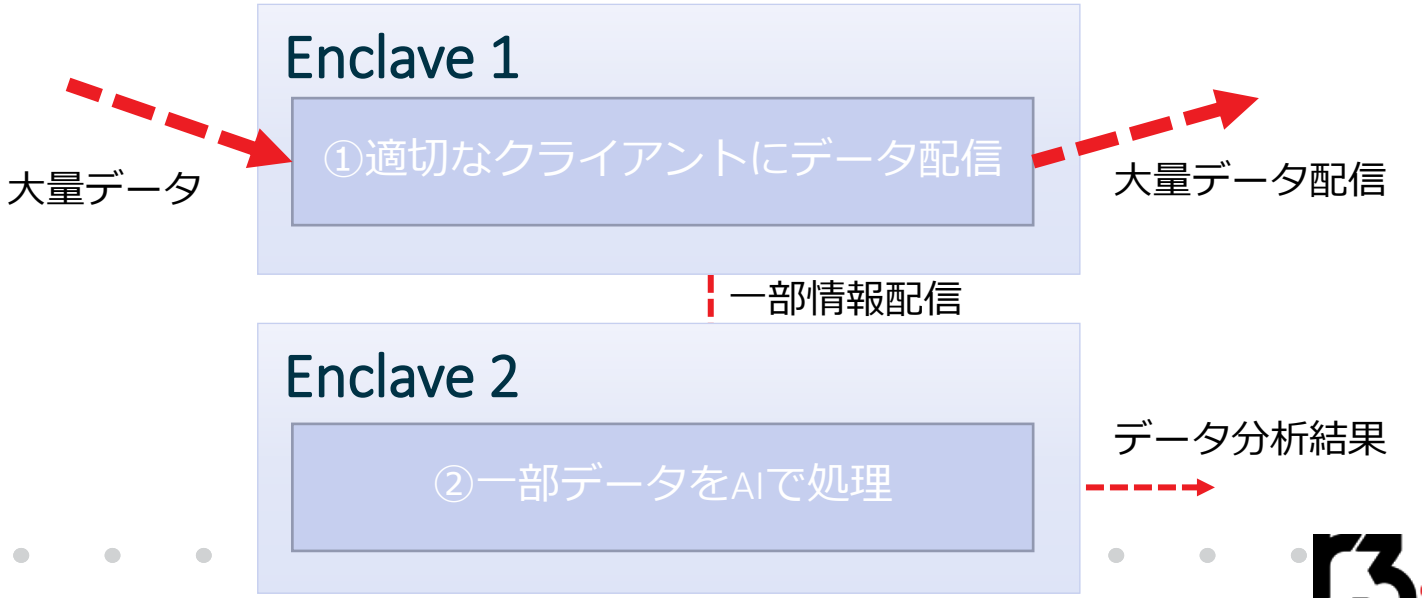
✓バックログをクリアできる保証はいる

制限事例③：大量処理：Enclaveの複数使用



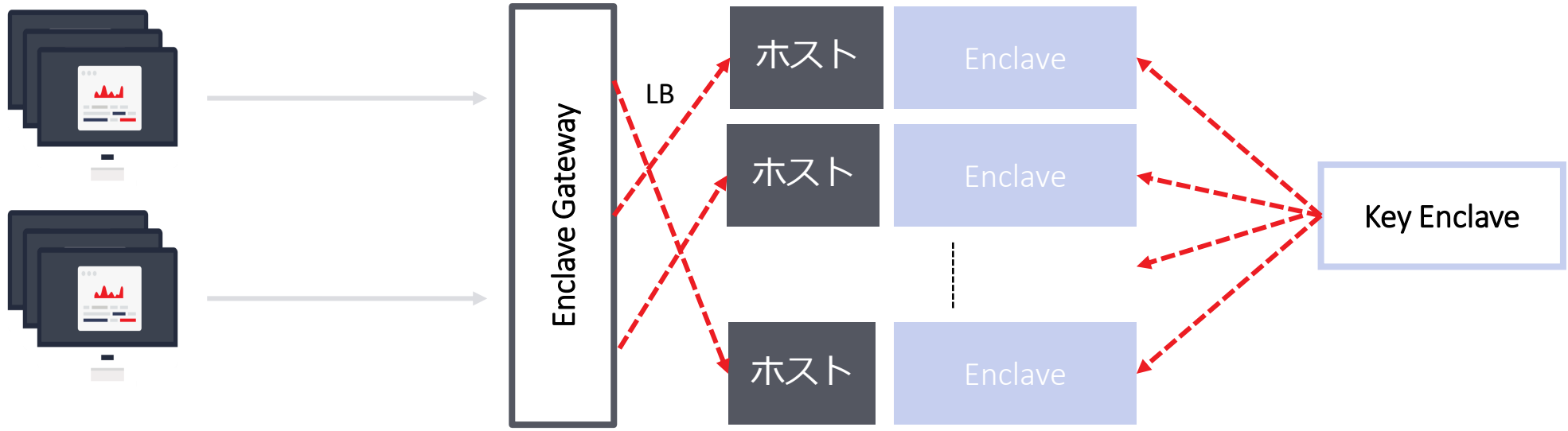
１つのEnclave

２つのEnclave



SBI R3 Japan

制限事例④：高可用性 or 大量処理（Enclave Pool）



Enclave Pool

✓複数のSGXシステム上のEnclaveに負荷が分散される

✓すべてのEnclaveが1つのデータストレージを共有

✓プール内のEnclaveが鍵を共有するための安全な方法が必要（Key Enclave）

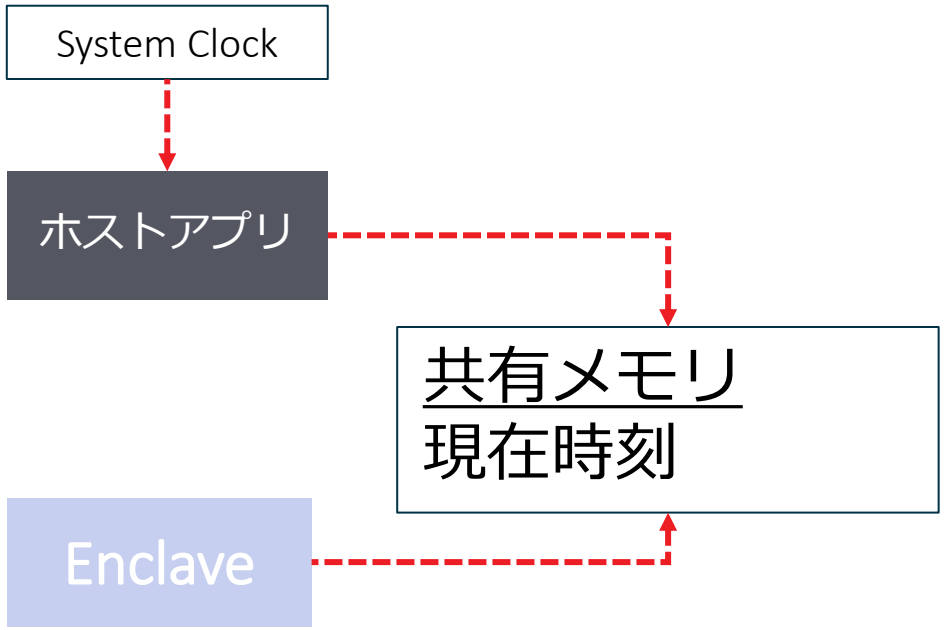
SBI R3 Japan

SBI R3 Japan

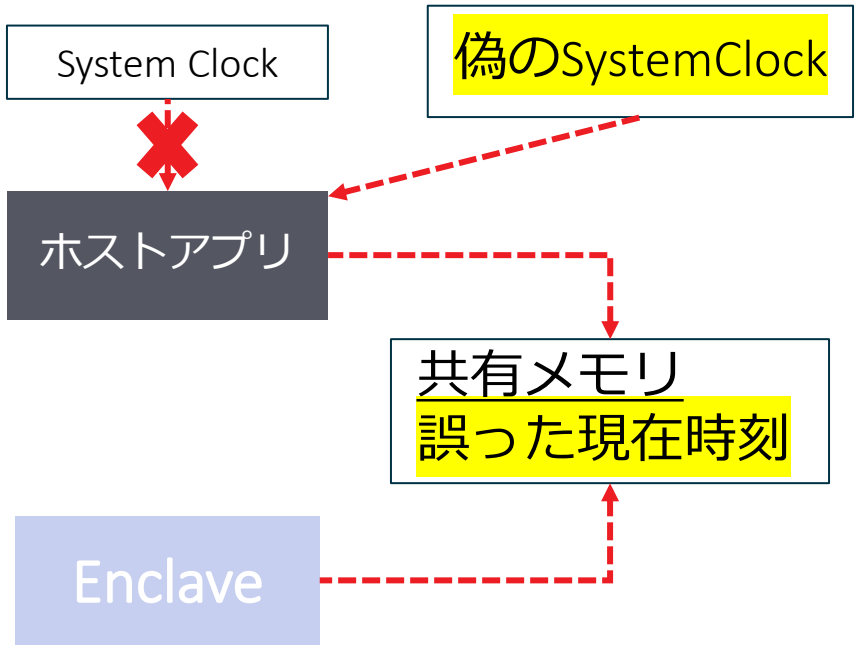
Security Consideration

アジェンダとの統合

Enclave使用時のセキュリティ課題①時刻

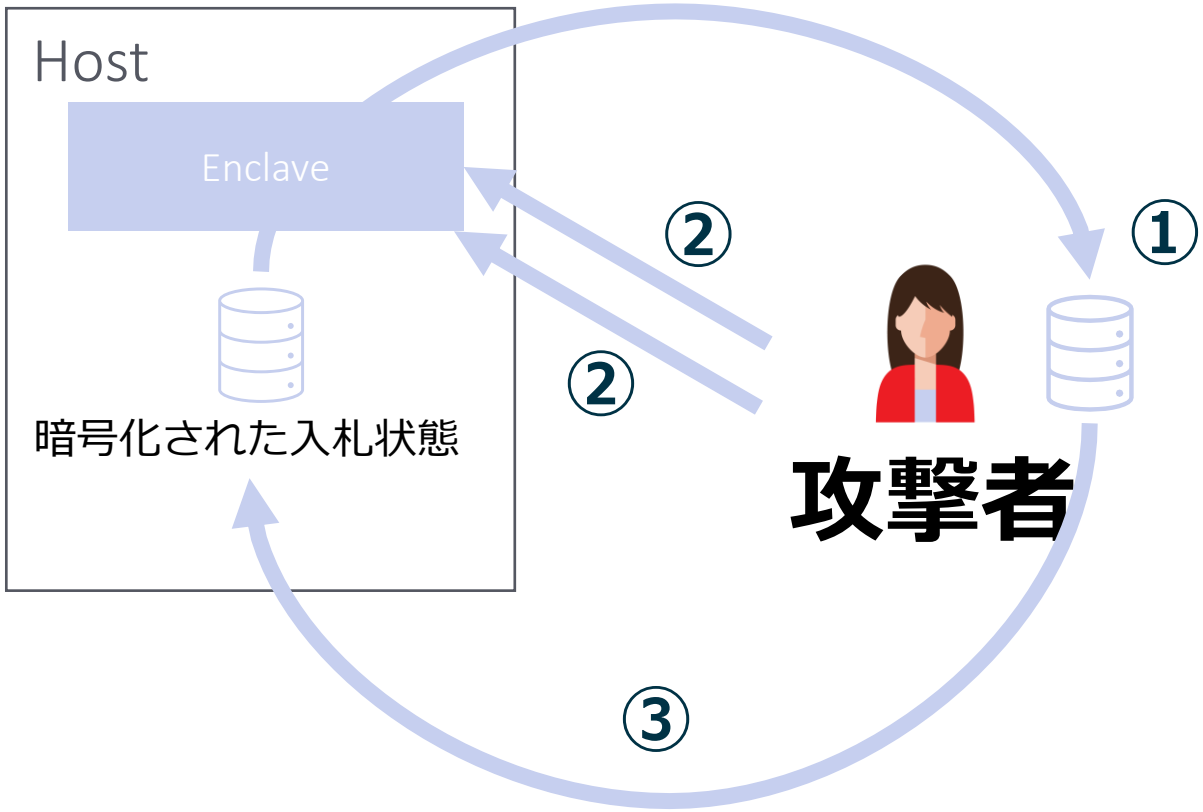


正しく動作するホスト



汚染されたホスト

Enclave使用時のセキュリティ課題②Replay 攻撃



（例）封印入札に対して

入札情報は暗号化して保存

① 攻撃者は入札情報をコピー

② 適当に入札する

1. 入札に負けた場合・・

③ 最初の入札情報をロードする

④ 違う値で入札する

Enclave使用時のセキュリティ課題③サイドチャネル攻撃

サイドチャネル攻撃

＝“操作の特徴を観察して、穴を見つける“

- ◆実行時間
- ◆メモリアクセスパターン
- ◆電力使用量

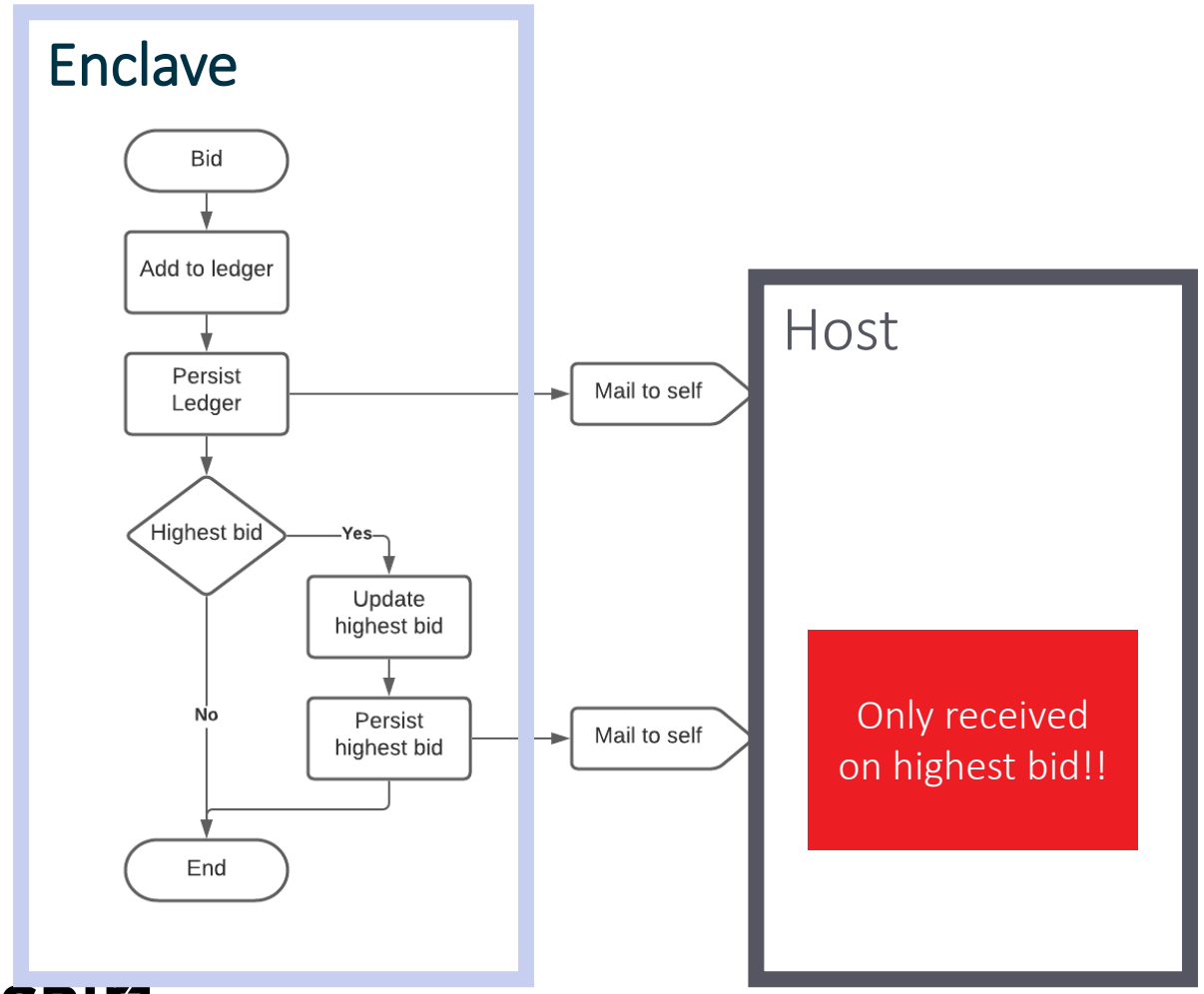
各緩和策(Mitigations)は、ほぼ自動的にConclaveにも適用される

From: https://arxiv.org/pdf/2006.13598.pdf

Attacks (abbreviated titles)	Type	SGX Specific	Targeted attack	Impact: Page access pattern	Impact: Instruction trace	Impact: Instruction latency	Impact: Memory access pattern	Impact: Memory Contents	Impact: Fault Injection	Mitigations: Microcode patch	Mitigations: System design	Mitigations: Compiler/SDK	Mitigations: Application design
Controlled-Channel [9]	sec. III-A	●	●	●	○	○	○	○	○	○	○	●[52]	●[55]
Stealthy Page Table [10]	sec. III-A	●	●	●	○	○	○	○	○	○	○	●[52]	○
SGX-Step [11]	sec. III-A	●	○	●	●	●	○	○	○	○	○	●[11][48]	○
Nemesis [12]	sec. III-A	●	●	○	○	●	○	○	○	○	○	●[50]	○
Off Limits [13]	sec. III-A	●	●	●	◐	○	◐	○	○	●[13]	○	○	●[13]
Leaky Cauldron [14]	sec. III-B	●	●	●	○	○	●	○	○	○	○	●[50][51]	○
CacheZoom [20]	sec. III-B	●	●	○	○	○	●	○	○	○	○	●[50][51]	●[56]
Cache Attacks on SGX [21]	sec. III-B	●	●	○	○	○	●	○	○	○	○	○	●[56]
Malware Guard Extensions [22]	sec. III-B	●	●	○	○	○	●	○	○	○	●[22]	○	○
Software Grand Exposure [23]	sec. III-B	●	●	○	○	○	●	○	○	○	○	○	●[23]
CacheQuote [24]	sec. III-B	●	●	○	○	○	●	○	○	○	●[24]	○	○
MemJam [25]	sec. III-B	○	●	○	○	○	●	○	○	○	○	●[49]	○
Branch Shadowing [26]	sec. III-C	●	●	○	●	○	○	○	○	○	○	●[26][48]	○
BranchScope [27]	sec. III-C	○	●	○	●	○	○	○	○	○	○	●[27][48]	○
Bluethunder [28]	sec. III-C	●	●	○	●	○	○	○	○	○	○	●[28]	○
SgxPectre [31]	sec. III-D	●	●	○	○	○	○	●	○	●[47]	○	○	○
SpectreRSB [32]	sec. III-D	○	●	○	○	○	○	●	○	○	○	○	●[32]
Spectre v1 [33]	sec. III-D	○	●	○	○	○	○	●	○	●[47]	○	○	○
Foreshadow-SGX [34]	sec. III-E	●	○	○	○	○	○	●	○	●[35]	○	○	○
RIDL [38]	sec. III-F	○	●	○	○	○	○	●	○	●[38]	○	○	○
ZombieLoad [39]	sec. III-F	○	●	○	○	○	○	●	○	●[39]	○	○	○
CacheOut [40]	sec. III-F	●	●	○	○	○	○	●	○	◐[40]	○	○	○
CrossTalk [41]	sec. III-F	●	○	○	○	○	○	●	○	●[41]	○	○	○
Plundervolt [44]	sec. III-G	●	◐	○	○	○	○	○	●	●[44]	○	○	○

SBI R3 Japan

Enclave使用時のセキュリティ課題④開発ベンダー由来のサイドチャネル攻撃



単純な例（封印入札）

- 入札時にホストへメール送信
- 最高額入札時にメール送信

⇒

- 最高入札時を知ることができる

SBI R3 Japan

Future

実装予定の機能ご紹介

Conclave: 今後実装予定の機能（確定ではない）

- **セキュリティ関連**
  - サイドチャネル攻撃緩和策の継続的取り込み
- **UX（エンジニアにとっての・・・）**
  - 共通デザインパターン: REST 利用の為のテンプレート, RPC, etc.
- **開発ライブラリ提供**
  - 機械学習用DB
  - 検索SDK
- **運用**
  - マネージドサービス

SBI R3 Japan