Vol.2021-CSEC-94 No.29
Vol.2021-SPT-43 No.29
2021/7/20

サーバレスアプリケーションにおけるデータの変化に伴う機密度再計算手法の提案

田村　悠[1]　磯部　義明[1]

概要：近年，COVID-19 によるテレワークの増加などにより，業務アプリケーションのクラウド移行に対する需要が高まっている．特にデータ駆動のビジネス判断にも用いられる分析業務において，迅速，柔軟かつ容易に分析プログラムが継続的に更新可能な DevOps 的手法としてサーバレスアプリケーションでの実装が普及し始めている．しかし，DevOps 的運用開発においては，継続的なセキュリティ設計・運用に課題がある．特にデータの保護に関しては GDPR などの観点からデータの性質に沿った適切な管理が必要とされている．しかし，サーバレスアプリケーションでは，内部で行われる処理の過程でデータの性質が変化するため，入力データの性質を基に適切な保管先を決定することができない．そこで本稿では，データに対し，そのデータの機密性や個人特定性などのセキュリティに関する属性である「機密度」を割当て，これをサーバレスアプリケーションにおける処理毎に再計算することで，データの保管時にその性質を基にして保管場所を決定することのできる手法を提案する．本手法を基に実装したシステムを利用した結果，個々のデータが適切な保管場所に保管されることを確認した．

Proposal of method to recalculation confidentiality cause to change data for serverless application

Yu Tamura[1]　Yoshiaki Isobe[1]

Abstract: Nowadays, demand of migrating business applications to cloud is increasing due to the increase in telework because of COVID-19. Especially in analytical work, migrate using serverless method is being to spread because the analysis program can be continuously updated like DevOps. However, this solution has problems in continuous security design and operation. Especially in data security, GDPR requires proper management according to the data property. However, it is difficult to decide appropriate storage for data using input data's properties because data and data properties change due to operation in serverless application. These services cannot store data to appropriate storage. We proposed the method that tracks how input data is modified by each process in a server-less application and recalculates its confidentiality accordingly, making it possible to select an appropriate storage. We confirmed in the experiment all data is stored in appropriate storage using this method.

1. はじめに

近年，Society5.0 の実現や，COVID-19 によるテレワークの増加などにより，社内のオンプレミスシステムとして構築されていた業務アプリケーションのクラウド移行に対する需要が高まっている．特に，IoT 機器やセンサなどから取得された大量のデータの分析においてサーバレスアプリケーションでの DevOps な運用実装が普及し始めている．サーバレスアプリケーションでの実装運用においては，インフラの構築・維持管理が不要なため，アプリケーションの開発に集中することが可能である．これにより，大量のデータの，ビジネスに応じた様々な分析を迅速かつ柔軟に実施できるようになった．

サーバレスアプリケーションを業務アプリケーションとして使う上で，セキュリティの問題は重要なポイントの一つである．特に，データ分析アプリケーションでは顧客の

[1] (株) 日立製作所
Hitachi Ltd.

購買履歴等の個人情報や，自社他社問わず企業の機密情報を取り扱う可能性が高いため，これらの情報の適切かつ安全な利用・保管が求められる．海外においても，個人情報については EU の GDPR[1] や NIST SP800-171[2] を始めとしたさまざまな法律，規則などで厳格かつ適切な保管が要求されている．しかし，サーバレスアプリケーションを用いたデータの適切な取扱は，開発時・運用時に考慮する事項が増えるため，前述の開発の柔軟性・迅速性に反することとなる．万が一開発の柔軟性・迅速性のみを目的として疎かにすると，結果的にセキュリティインシデントを起こすこととなり，企業は社会的・経済的に大きなダメージを負うことになりかねない [3]．また同時に，セキュリティの為に前述の柔軟性・迅速性を放棄するとデータ分析による迅速なビジネス判断に影響が発生する．

本研究では，アプリケーションに入力されるデータに「機密度」と呼ばれる，そのデータの機密性や個人特定性などのセキュリティに関する属性を割当てる手法を提案する．本手法をサーバレスアプリケーションに対し，アプリケーション内の処理毎に機密度の再計算を実施する形で適用することで，データの保管時にその性質を基にして保管場所を決定することが可能となる．

以下，まず 2 章においてサーバレスアプリケーションにおけるデータセキュリティ上の課題を述べる．次に 3 章で提案手法を述べ，4 章で実装及び評価について示す．5 章で考察を述べ，最後に 6 章でまとめと今後の課題を述べる．

2. サーバレスアプリケーションにおけるデータセキュリティ上の課題

2.1 データセキュリティの検討事項

サーバレスアプリケーションにおけるデータセキュリティに際して必要な検討事項は，主にデータの適切な保管先に関するものである．アプリケーションに対して入力されたデータは，アプリケーション内で様々な処理を経て，ストレージに保管される．この時，入力データと出力データでは内部の処理を経ることでデータ及びその性質が変化しており，そのデータの性質に応じた適切なストレージへの保管がデータセキュリティの実現には要求される．しかし，データの保管処理も関数の 1 つとして実装する必要があるサーバレスアプリケーションにおいてこの実装を要求することは，投入されるデータとそのデータが通る経路毎に専用のデータ保管関数を用意することを示している．特に，複数の性質の異なるデータが同一の経路を通ることになった場合，出力データの内容に基づいてこの判定を要求され，これらの検討やテストは時間的および人的コストがかかる．また，これらはアプリケーションの構成が組み変わるたび，また各関数の実装が変わるたびに検討する必要があり，これは迅速かつ柔軟なアプリケーション設計・構築の要件を満たさない．データセキュリティを実現するために迅速さ・柔軟さを放棄することはサーバレスアプリケーションでの構築をするメリットを失うものであり，このメリットを残したままデータセキュリティを確保する方法が求められる．

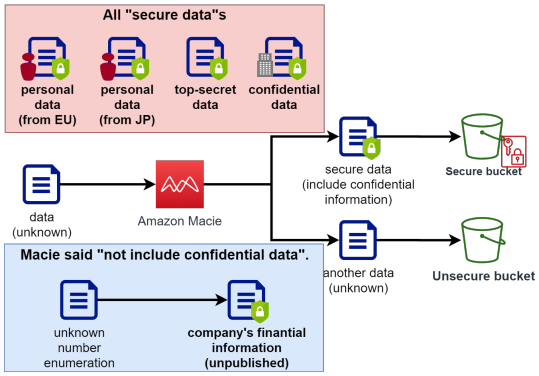


図 1 Amazon Macie の概要
Fig. 1 System of Amazon Macie

2.2 既存手法の例とその課題

サーバレスアプリケーションにおけるデータセキュリティを実現する既存手法として，クラウドサービスプロバイダが提供する機密情報検知サービス [4][5][6] である．これは，各クラウドサービスのオブジェクトストレージに保管されたデータを定期的にスキャンし，機密情報を検知するサービスである．例としてここでは AWS(Amazon Web Services) の提供する Amazon Macie[4] について述べる．Amazon Macie では，S3(Simple Storage Service)[7] に保管されたデータをスキャンし，機密情報を検知する．検知方法は個人情報に関しては機械学習を，それ以外については利用者が設定したルールに基づくパターンマッチングを使用する．また，検知結果に基づいたアクションを設定することが可能で，このアクションを利用することで当該データの別ストレージへの移動が可能である．これにより，機密情報とそれ以外を分けて管理することが可能である (**図 1**)．また，このサービスはストレージ上のデータに対して提供するものであり，データを保管するアプリケーション側での操作が一切不要なため，アプリケーション開発者が意識することなく，データのセキュリティを確保することができるとされている．

しかし，当該サービスではあくまで機密情報の有無を検知，移動することのみが可能となっており，そのデータがどのような機密情報であるかまでは判定ができないという問題がある．特に個人情報については，EU の GDPR[1] を始めとして各国でその取扱方法に差があり，グローバルなサービスを提供する場合はその利用者の属性やサービス提供国などによってデータの取扱を分ける必要がある．それ以外の機密情報についても，ごく限られた関係者にのみしか開示できない情報もあれば，組織に所属さえしていれば

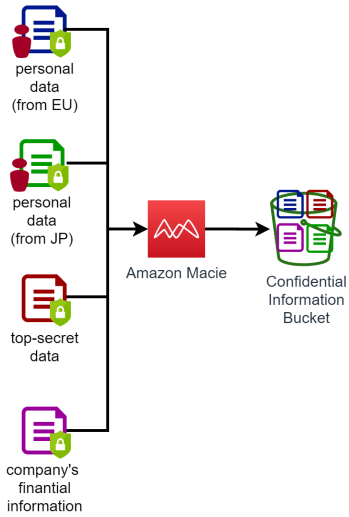


図 2 Amazon Macie よるデータ制御の概念図
Fig. 2 Abstruction of Amazon Macie Data Control.

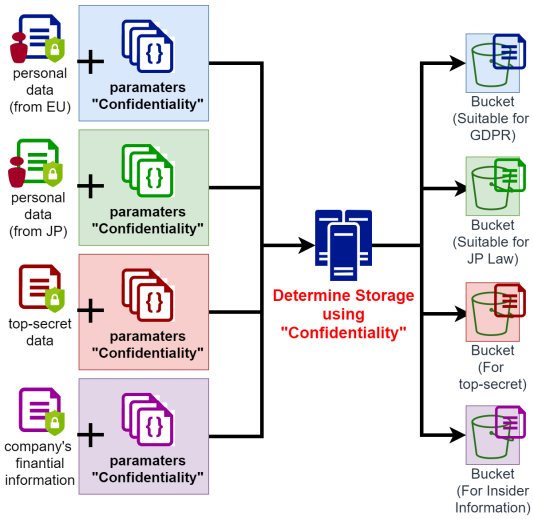


図 3 機密度によるデータ制御の概念図
Fig. 3 Abstruction of Confidentiality based Data Control.

誰でも閲覧可能な情報もあり，これらを単一の設定のストレージで管理するのは現実的ではない．そのため，Amazon Macie の機能では，機密情報の隔離はできても，適切な管理を実現することは難しいという問題がある．

3. 提案手法

3.1 提案手法の概要

本提案手法は，データのセキュリティに関する性質を示す指標として「機密度」を導入し，機密度の値によってデータを制御するものである．

機密度はデータを制御する上で必要なプライバシーや機密性などのセキュリティに関する属性を多角的に評価し，評価した属性毎にその結果を数値化した値の集合である．なお，評価項目は取り扱うデータや組織が順守すべき法規等を基準に予め定めるものとし，本論では当該評価項目や評価項目における値の算出方法については述べない．

以下，機密度を用いたデータ制御について 3.2，サーバレスアプリケーションにおける活用方法について 3.3 以降で述べる．

3.2 機密度を用いたデータ制御

機密度を用いることで，様々な性質を持つデータについて，適切なストレージに保管することが可能となる．例を図に示す．**図 2** が既存手法，**図 3** が提案手法に置けるデータ制御の考え方である．

既存手法である Amazon Macie ではデータから機密情報と判定されるものが発見された場合，特定の機密情報用ストレージに移動することが可能である．この手法による管理が不適切であることは前に述べた通りである．

ここで，提案手法では各データに対して，そのデータの性質を示した「機密度」を割り当てる．機密度はセキュリティに関する性質を複数の評価軸で，評価軸毎に数値化したものの集合であるため，「機密情報であるか否か」以外に多数の情報を含むことができる．この例では，以下のような情報が機密度として数値で表現される．

- そのデータは法規等での規制の対象となる個人情報に該当しうるか
- そのデータの組織に置ける機密性の度合いはどの程度か
- 個人情報である場合，GDPR の規制対象になるか
- 個人情報である場合，日本の個人情報保護法及び関連諸法に依る規制対象かどうか

これらの情報を基に，機密度に関する判定式を予め設定しておくことにより，データをその保管方法に応じて適切に分離することが可能となる．同時に，それぞれの判定結果に適合した適切なストレージを予め用意しておくことで，それらのデータは対応する適切なストレージへと保管することが可能となる．

これによって，以下の要件を満たす場合において，本提案手法は様々な性質を持つデータをその性質に適したストレージへ振り分けることが可能となる．

- 取扱い得るすべてのデータについて，適切な保管方法を把握し，当該設定を施したストレージ（オブジェクトストレージサービス等）を用意する
- データを保管する上で判断すべき性質を機密度として定義する
- 保管場所を決定するための機密度の判定式を予め定義する
- 各データについて，機密度を適切に割り当てる

3.3 サーバーレスアプリケーションにおける活用方法

本提案手法をサーバレスアプリケーションにおいて活用する上では以下について検討する必要がある．

- 機密度の割り当て方法

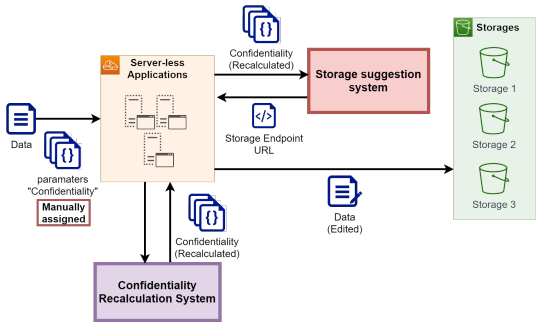


図 4 サーバレスアプリケーションにおける機密度を用いたデータ制御の構成図

Fig. 4 Structure of Data Control using Confidentiality on Serverless Application.

- アプリケーション開発の障害とならない構成

1 つ目の機密度の割当方法について，現状機密度をデータから自動で算出することは難しいため，何らかの方法で保管時の機密度を算出する必要がある．特にデータ分析アプリケーションにおいては，アプリケーションに入力したデータに様々な加工を施した後にデータの保管が行われるため，入力時のデータと保管時のデータでセキュリティの性質，すなわち機密度が変化している事に留意する必要がある．

2 つ目について，2 章で述べた通り，迅速かつ柔軟なアプリケーションの設計・構築が可能であるというサーバレスアプリケーションのメリットを残したまま提案手法を適用する必要がある．

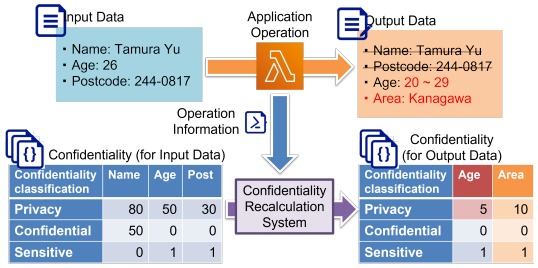
上記について，本論では**図 4** に示す構成で解決した．

本構成は，入力データに割り当てられた機密度を，アプリケーション内の各処理で再計算することで保管時の機密度を算出し，その機密度を用いて保管場所を判定し，ストレージに保管するものである．機密度再計算システム (Confidentiality Recalculation System) は機密度の再計算を行うコンポーネントである．また，ストレージ提案システム (Storage Suggestion System) は機密度から保管場所を判定し，アプリケーションに対して提案するシステムである．

本構成は以下のステップで動作することで，提案手法の適用を実現する．

(1) サーバレスアプリケーションに入力されるデータ及び内部で使用されるデータについて，手動，ないしは任意のツール等を用いて機密度を割り当てる．
(2) アプリケーションにデータを入力する時にデータと同時に当該機密度を入力する．
(3) アプリケーション内の 1 番目の処理 (関数) が実行される
(4) 処理後，次の処理 (関数) に行く前に機密度再計算の機能で当該処理による機密度の変化を再計算する
(5) 処理の結果と機密度再計算の結果を 2 番目の処理 (関数) に送る
(6) 3,4 を最後の処理まで繰り返す
(7) 全ての処理が終わり，データを保管する段階に入ったら最後の機密度計算の結果を基にストレージ提案の機能で保管場所を指定する
(8) 7 で指定されたストレージにデータを保管する
(9) アプリケーションの処理結果を返答する

以下，機密度再計算システムについてを 3.4，ストレージ提案システムについて 3.5 で述べる．



Confidentiality classification	Name	Age	Post
Privacy	80	50	30
Confidential	50	0	0
Sensitive	0	1	1

Confidentiality classification	Age	Area
Privacy	5	10
Confidential	0	0
Sensitive	1	1

図 5 機密度再計算の例

Fig. 5 Example of Confidentiality Recalculation.

3.4 機密度再計算

本機能では，サーバレスアプリケーションを構成する各処理（関数）単体においては，自身の処理がデータにどのような影響を与えることが容易に判別可能であることを利用して，当該処理単体における「機密度」の変化を計算する．入力データ・使用した内部データ・出力データの項目リスト，入力データ・使用したデータの機密度リスト，実施した処理に関する情報を入力することで，出力データの機密度リストを出力する．以下，具体例を述べる．

図 5 は機密度再計算の具体例を示した図である．この例では機密度の分類として以下の 3 種類を定義した．

- Privacy：個人特定度 (0-100)
- Confidentiality：組織におけるデータの機密性 (0-100)
- Sensitive：取扱に注意が必要なセンシティブ情報の有無 (0(無) or 1(有))

この処理は，名前 (Name)，年齢 (Age)，郵便番号 (Postcode) の入力を年代 (Age) と居住都道府県 (Area) に変換することで，データの匿名化を実施するものである．機密度の視点でこの処理を見ると，この処理によってデータ全体の Privacy 値や Confidentiality 値が下がるはずである．

まず，処理に先立って入力データの Name，Age，Postcode の機密度を，上記基準に基づき図 5 左下の表の通りに設定した．この機密度の値は，データの以下の性質を表現している．

- Name：同姓同名がいる可能性はあるものの，ほぼ一意に一個人を特定できるため，Privacy 値が非常に高い．また，Confidentiality 値も高い．
- Age：年齢単体はただの数字のため，Confidentiality 値

は 0 であるものの，Privacy の観点でみると他の情報と結合した際に個人を特定しやすくなるため，Privacy 値が比較的高めである．

- Postcode：Confidentiality 値は Age と同様の理由で 0 となる．Privacy の観点では同様に他の情報と結合した際の個人特定度が多少あるものの，市区町村レベルまでしか特定できないため，Age よりは特定度が低い．

処理後のデータの機密度は，図 5 右下となる．この機密度は，それぞれ年代・居住都道府県という形で抽象化されており，他の個人特定度が低めのデータと結合しても個人を特定できる可能性が低いという性質を表している．また，年代と都道府県では，値のパターンが都道府県の方が多いため，Area の方が Privacy 値が多少高くなっている．

前述の条件の元，図 5 の動作のステップを以下に示す．

(1) アプリケーションの処理の中で，図 5 の処理が呼び出される．
(2) 処理を実行し，出力データを生成する
(3) 機密度再計算機能に，以下のデータを送る
- 入力データの項目：Name, Age, Postcode
- 出力データの項目：Age, Area
- 入力データの機密度：図 5 左下の表の通り
- 処理に関する情報 1：Name を削除した
- 処理に関する情報 2：Postcode から Area を生成した，機密度は図 5 の Area 列の通り
- 処理に関する情報 3：Age の情報を変更した，機密度のうち，Privacy が 5 に変化した

(4) 機密度再計算機能は送られてきたデータを基に，出力データの機密度である図 5 右下の表を生成する
(5) 機密度再計算機能は出力データの機密度を処理に対して返答する
(6) 次の処理に対して出力データと出力データの機密度を送信する

以上の例のような操作を，アプリケーションにおける各処理の後に呼び出し，実施する．

本提案においては，機密度再計算システムはアプリケーション内の各処理 (関数) と同様の関数の 1 つ，もしくは RESTful な API として定義することを想定している．そのため，機密度再計算システムは関数の処理の最後，次の処理にデータを送る前に実行するコードを挿入する．これらは関数を組み替えても通常，正常に動作し，また各処理の修正時にも大きな影響を及ぼさない．これにより，アプリケーションと疎結合に本機能を実現可能である．

3.5 ストレージ提案

ストレージ提案は 3.4 で算出された機密度を用いて，データ保管前にそのデータが保管されるべき適切なストレージを提案する機能である．

本機能は，予め管理者により設定された以下の情報を用いて，機密度から保管先ストレージを判定する．

- データの分類を機密度から判定するための判定式
- データの分類とマッピングされるストレージの宛先

本機能も 3.4 と同様に，RESTful API として定義することを想定している．そのため，アプリケーション側は機密度を本機能に送るだけで，適切な保管先ストレージを取得することが可能である．これにより，アプリケーション実装においてはデータのセキュリティ上の性質や保管先について開発者が意識する必要がなくなる．

また，本機能には前述の 2 つの情報を設定する機能も持つことを想定している．これにより，データの保管先については，アプリケーション開発と分離して，セキュリティの専門家や管理者によって独立して指定することが可能であり，アプリケーション開発のフローを乱すことなく，データセキュリティを確保することが可能である．

4. 実装・評価

4.1 評価手法

提案手法の実装可能性と適切な動作を確認するため，提案手法と，提案手法を適用する実験用業務アプリケーションを AWS 上に実装し，その動作を確認した．

本実験において評価する項目は，(1) 実験用業務アプリケーションが適切に動作する事，(2) アプリケーション側に特段の設定をすることなく，異なる機密性を持つ複数のファイルがそれぞれ意図したストレージに保管される事，の 2 点である．

また，参考情報として，提案手法利用時とそうでない場合での処理時間の違いを測定した．これは，処理時間に大きな差が発生した際，実装について再検討するためである．なお，アプリケーションの用途により許容遅延時間は異なるため，本稿ではこのデータを用いた実行時間の妥当性については評価を実施しない．

4.2 実験用業務アプリケーションの実装

実験用業務アプリケーションとして，自動車 IoT データの分析システムを実装した．**図 6** に概念図を示す．

この業務アプリケーションは，自動車メーカーが利用することを想定したものである．自動車に搭載された IoT モジュールから毎分センサデータや走行データなどが送られてくる状況において，以下の 5 種類のデータを生成する（カッコ内は機密度合の種別）．

- 生データ（個人情報）
- 車種毎の診断情報：車種に共通する不具合がないか分析するためのデータ（社外秘データ）
- GPS データの集合：渋滞情報の生成のために使われるデータ（公開データ）
- 顧客別利用状況分析データ：車両の利用状況などから顧客に新規サービスなどを提案するためのデータ（個

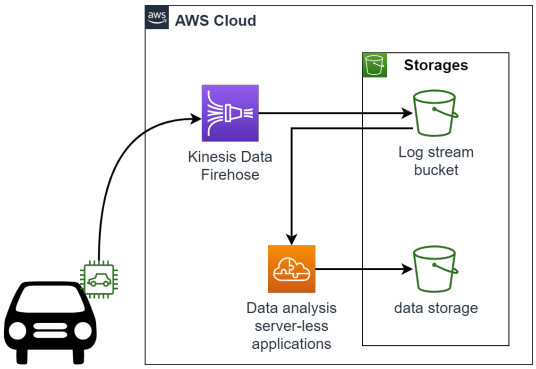


図 6 実験用業務アプリケーションの概念図
Fig. 6 Abstruction of Experimental System.

人情報）

- 車種別利用状況分析データ：車種毎に利用状況をまとめ，当該車種を利用している顧客の利用状況を分析することで販売戦略等を検討するためのデータ（マル秘データ）

本実験では上記で表した 4 種別の他に極秘データ用のストレージの 5 種類を用意し，データがすべて適切な種別のストレージに保管されるかどうかを確認した．

4.3 実装

実装の構成を**図 7** に示す．本実装では，自動車から送られてきたデータを一時的に保管するバケットにデータが入ると，CloudWatch Event で DataSplitterEvent が実行されるように Lambda トリガーを設定した．また，自動車 IoT から一時バケットまでの部分は EC2 上で動作する Python スクリプトで動作を模擬した．提案手法は API Gateway と Lambda を用いて単一の API として実装した．また，実験用業務アプリケーションの各関数において処理の最後に機密度再計算機能を，データ保管時にストレージ提案機能を呼び出すよう実装した．また，5 種類のストレージはそれぞれ性質に応じて異なる公開範囲，アクセス管理，ログ，暗号化の有無，暗号化方法などの設定で構築した．

4.4 評価

4.3 の実装で，EC2 上の Python コードを 1 分に 1 回，合計 60 回呼び出してその動作を確認した．結果，すべてのデータが適切なストレージに保管されることを確認した．これにより，提案手法は 4.2 の実験用業務アプリケーションにおいて，適切に動作することが確認された．

5. 考察

本手法によって，少なくとも今回実装した実験用業務アプリケーションにおいては適切にデータが保管されることを確認した．これにより，以下の条件を満たす場合は本手法は適用可能であると言える．

- 入力データの機密度が任意の方法で算出済みである
- 各処理 (関数) において，当該処理がどのデータ項目に対してどの程度機密度に影響が発生するかが明示的に示されている

そのため，本手法をそのまま適用する場合は，入力データに対して設定する機密度の値の妥当性，及び各関数で指定する機密度の変化値の妥当性を別途任意の手段で検証する必要があると考えられる．

また，そもそも前述の 2 条件が現実的でないという指摘も考えられる．特に，機密度の分類とその値の設定方法について，誰が設定しても同じデータについては同じ機密度が設定されるようなポリシーは非現実的であるという点については否定できない．これについては，データから自動的に機密度を算出する仕組みや，機密度の入力をデータと同時ではなく，データ項目名に対して紐づけることで，予めセキュリティの専門家によって適切な値を設定させる，などが考えられる．しかし，前者は実現可能性について十分な議論が必要であること，後者については開発者に対して特別な制約を与えることになるため，引き続き検討が必要である．

さらに，本手法ではストレージのセキュリティ設計については引き続き手動で実施する必要がある．しかし，各データに適したストレージの設定，特に個人情報等の法律等で管理方法が定められているものについては，それらに精通した者が設計をする必要がある．この点においては，当初の課題が解決されていない．本項目については，ポリシーを Machine Readable な形にすることで，ストレージの設定を自動生成することが可能であると考えている．クラウドにおけるコストなどの制約要件に基づいたリソースの自動生成は [8][9][10] など各種先行研究があり，これらと組み合わせることにより実現可能であると考えられる．

6. まとめと今後の課題

本稿では，サーバレスアプリケーションにおけるデータの適正な保管を，アプリケーションの実装に依存せずに実現するため，機密度とアプリケーションの処理毎に行われる機密度の再計算からデータの保管場所を導出する手法を提案した．本提案手法の実装可能性と適切な動作を確認するため，提案手法と実験用サーバレスアプリケーションを実装し，その動作を確認した．結果，すべてのデータが適切な場所に保管されることを確認した．

本手法の課題として，機密度算出手法の定量化又は自動化，ポリシーに基づいたストレージの自動設計，及び複数の実際に稼働するサーバレスアプリケーションにおける実証などがある．

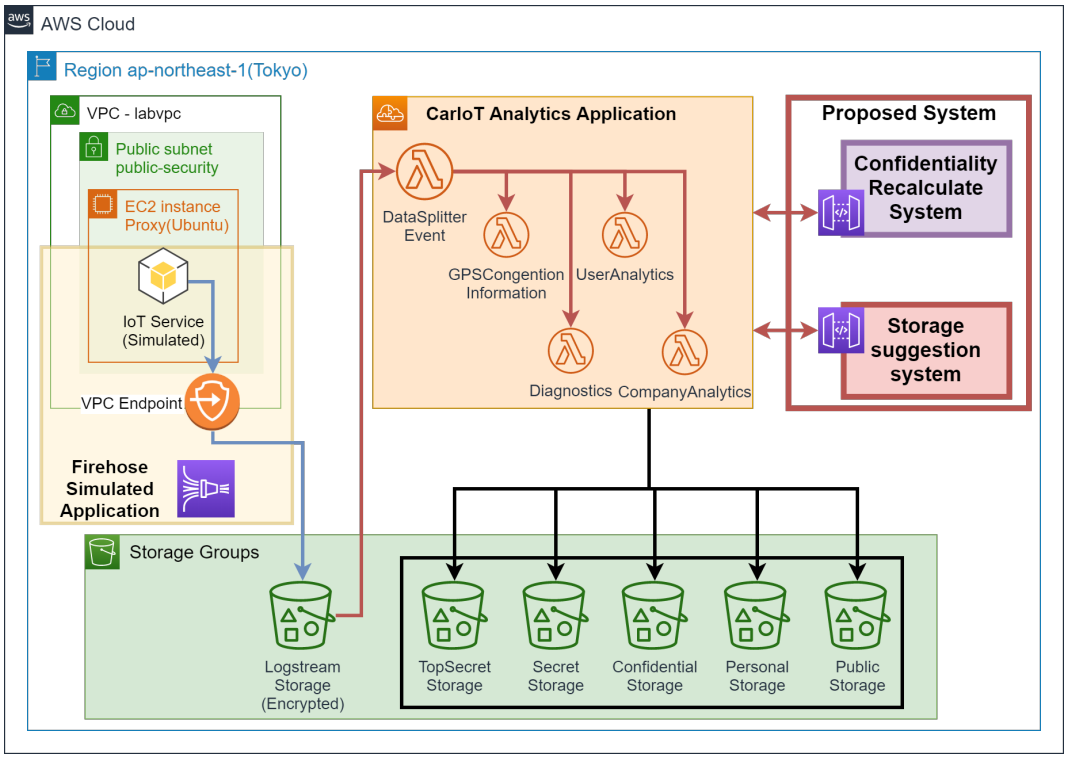


図 7 実装構成図
Fig. 7 Impremention.

るその他の AWS 商標は，米国およびその他の諸国における，Amazon.com, Inc. またはその関連会社の登録商標または商標である．Python は，米国およびその他の諸国における，Python Software Foundation の商標または登録商標である．本稿に記載されているその他の会社名，製品名は，それぞれの会社の登録商標もしくは商標である．

参考文献

[1] Europian Union: *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*, 入手先 ⟨https://eur-lex.europa.eu/eli/reg/2016/679/oj⟩, EU(2016)
[2] National Institute of Standards and Technology: *SP 800-171 Rev. 2 Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations*, 入手先 ⟨https://doi.org/10.6028/NIST.SP.800-171r2⟩, NIST(2021)
[3] インフォマティカ・ジャパン：データプライバシーの新たなパラダイム～変化する脅威に対応するための持続的アプローチ, 入手先 ⟨https://blogs.informatica.com/jp/2020/01/23/dataprivacy-paradime/⟩ (2020.11.09)
[4] Amazon Web Services, Inc.：Amazon Macie, 入手先 ⟨https://aws.amazon.com/jp/macie/⟩ (2020.11.09)
[5] Microsoft Corporation：Azure Information Protection, 入手先 ⟨https://azure.microsoft.com/ja-jp/services/information-protection/⟩ (2020.11.09)
[6] Google, Inc.：Cloud Data Loss Prevention, 入手先 ⟨https://cloud.google.com/dlp?hl=ja⟩ (2020.11.09)
[7] Amazon Web Services, Inc.：Amazon S3, 入手先 ⟨https://aws.amazon.com/jp/s3/⟩ (2020.11.09)
[8] Lama Palden. and Zhou Xiaobo.: *AROMA: automated resource allocation and configuration of mapreduce environment in the cloud*, ACM(2012)
[9] Mansouri Yaser. and Buyya Rajkumar：*To move or not to move: Cost optimization in a dual cloud-based storage architecture*, Journal of Network and Computer Applications, ELSEVIER(2016)
[10] Dutta S. Gera S. Verma A. and Viswanathen B: *SmartScale: Automatic Application Scaling in Enterprise Clouds*, IEEE(2012)