平成30年度　防災情報サービスプラット フォーム　　　　　　　　　の基盤整備支援業務

中核機能設計書

平成３０年１２月

独立行政法人防災科学技術研究所

目次

[1. 中核機能の要件定義 １](#_Toc532832886)

[1.1. 要件定義の進め方 １](#_Toc532832887)

[1.2. 中核機能の全体要件の定義 １](#_Toc532832888)

[1.2.1. 要件定義の要素 １](#_Toc532832889)

[1.2.2. データレイクの要件定義 ２](#_Toc532832890)

[1.2.3. イベントエンジンの全体要件定義 ４](#_Toc532832891)

[1.2.4. データディクショナリの全体要件定義 ５](#_Toc532832892)

[1.2.5. 全体要件定義の作業実施の内容 ７](#_Toc532832893)

[2. 中核機能個別の方式設計 ９](#_Toc532832894)

[2.1. データレイクの方式設計 ９](#_Toc532832895)

[(1) 多様なデータの種類や形式をサポートし、多様な操作を可能にする環境　及び多様なデータ形式をそれぞれの特性に合わせた格納を可能にする ９](#_Toc532832896)

[(2) セキュリティを担保し、データを連携させる範囲を設定できる　及び　自治体等のユーザーレベルでデータ独立性を担保したセキュリティの確保 １１](#_Toc532832897)

[(3) 自治体等のユーザーレベルで、データごとに共有先の設定を可能にする １３](#_Toc532832898)

[(4) データ形式や格納先を問わず、データの構造化・分析・集計のためのアクセスを可能にする １４](#_Toc532832899)

[(5) 災害インシデントの管理 １５](#_Toc532832900)

[2.2. イベントエンジンの方式設計 １７](#_Toc532832901)

[(1) イベントとしてデータの状態遷移を監視可能にする １７](#_Toc532832902)

[(2) イベントとして処理の状態遷移を監視可能にする １９](#_Toc532832903)

[(3) イベントとして時間の経過を監視可能にする ２０](#_Toc532832904)

[(4) エンジン機能として上記3 つの状態遷移を同じように処理できること ２１](#_Toc532832905)

[(5) エンジン機能として処理負荷に連動し、処理リソースの変化を可能にする ２２](#_Toc532832906)

[(6) エンジンとして処理の経過や状況を管理でき、使用者に提供を可能にする ２２](#_Toc532832907)

[(7) 処理の定義をある程度簡素化できる仕組みを考慮できること ２２](#_Toc532832908)

[2.3. データディクショナリの方式設計 ２３](#_Toc532832909)

[(1) ディクショナリとしてデータの形式を問わず、メタデータ（形式や属性情報）を登録可能にする ２３](#_Toc532832910)

[(2) ディクショナリとしてデータ種類毎のデータカタログへ登録する情報を登録可能にする ２４](#_Toc532832911)

[(3) ディクショナリとしてデータ種類毎の共有範囲を設定可能にする ２６](#_Toc532832912)

[(4) ディクショナリとしてデータ毎の管理機能（保管期間等）を提供できること ２６](#_Toc532832913)

[(5) 防災データ連携の標準化・高度化を目的に防災分野の語彙を災害対応業務の流れと併せて定義する ２６](#_Toc532832914)

[(6) 上記で作成する防災分野の語彙は共通語彙基盤と整合性をとること ２６](#_Toc532832915)

[(7) ディクショナリの定義をある程度簡素化できる仕組みを考慮できること ２７](#_Toc532832916)

[(8) 今年度設計のデータディクショナリのデータモデル ２７](#_Toc532832917)

[2.4. データ処理の設計 ２７](#_Toc532832918)

[2.4.1. 今期実装のデータ処理 ２７](#_Toc532832919)

[2.4.2. 今期実装のデータ処理の設計 ２８](#_Toc532832920)

[(1) 雨量データのGEOデータ化 ２８](#_Toc532832921)

[(2) 地震データのGEOデータ化 ２８](#_Toc532832922)

[(3) WebEOC（橿原市訓練データ）の連絡票テキストデータの自然言語処理 ２８](#_Toc532832923)

図表一覧

[図表 1‑1　中核機能への機能要求　抜粋(1) ７](#_Toc532832855)

[図表1‑2　中核機能への機能要求　抜粋(2) ７](#_Toc532832856)

[図表 2‑1　データレイクの構成 ９](#_Toc532832857)

[図表 2‑2　データレイクの想定格納データ形式 ９](#_Toc532832858)

[図表2‑3　データレイクの自治体が使用するサービス及び領域の独立概要 １１](#_Toc532832859)

[図表2‑4　データレイクのデータベースのユーザー概要 １２](#_Toc532832860)

[図表2‑5　データレイクのデータベースのユーザーとHDFS/オブジェクトストレージの独立性の関係 １２](#_Toc532832861)

[図表2‑6　29年度に検討された研究開発内容のデータ共有の概要 １３](#_Toc532832862)

[図表2‑7　本年度検討結果のデータ共有の概要 １４](#_Toc532832863)

[図表2‑8　データレイクの構成 １４](#_Toc532832864)

[図表2‑9　一元的な操作と独立性を担保したデータレイクの構成 １５](#_Toc532832865)

[図表2‑10　パーティションテーブルの構成 １５](#_Toc532832866)

[図表2‑11　 Event HUB機能を利用したメッセージをイベントとして扱い自動処理する方式 １７](#_Toc532832867)

[図表2‑12　 Event HUBのメッセージとイベントプロバイダーによる自動処理する方式 １８](#_Toc532832868)

[図表2‑12　 IoTのストリーミングデータのデータ連携方式 １８](#_Toc532832869)

[図表2‑14　 IoTのストリーミングデータの分析方式 １９](#_Toc532832870)

[図表2‑15　 Event HUBのイベントプロバイダーとDIPCによる自動処理する方式 ２０](#_Toc532832871)

[図表2‑16　 DIPCによるデータ処理の構成（接続構成） ２０](#_Toc532832872)

[図表2‑17　 DIPCによるデータ処理の構成（処理構成） ２０](#_Toc532832873)

[図表2‑18　 JOB ENGINEからEvent HUBのイベントプロバイダーへのメッセージ連携 ２１](#_Toc532832874)

[図表2‑18　 今期構成したイベントエンジンの方式 ２１](#_Toc532832875)

[図表2‑20　 DIPCのデザイン機能 ２３](#_Toc532832876)

[図表 2‑21　データレイクの想定格納データ形式 ２３](#_Toc532832877)

[図表 2‑22　メタデータ（形式や属性情報）の格納モデル ２４](#_Toc532832878)

[図表 2‑23　平成29年度に検討されたデータカタログ ２５](#_Toc532832879)

[図表 2‑23　データカタログのメタデータ収録部分 ２５](#_Toc532832880)

[図表 2‑25　データカタログの共有設定部分 ２６](#_Toc532832881)

[図表 2‑26　今年度設計したデータディクショナリのデータモデル ２７](#_Toc532832882)

[図表 2‑27　雨量データのデータ処理 ２８](#_Toc532832883)

[図表 2‑28　地震データのデータ処理 ２８](#_Toc532832884)

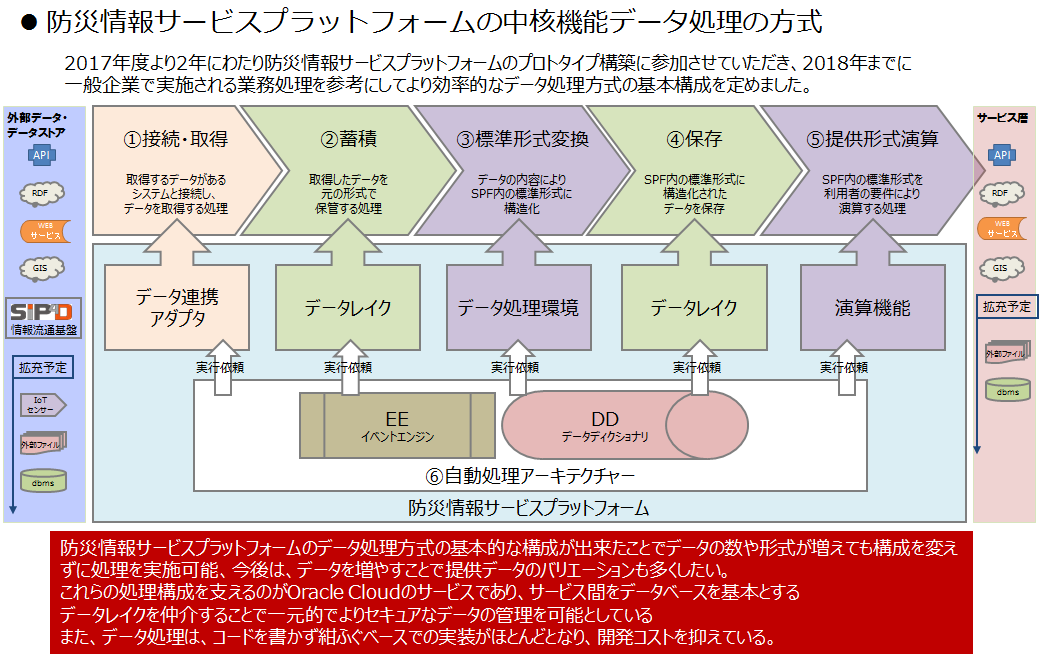
[図表 2‑29　自然言語処理解析結果 ２９](#_Toc532832885)

# 防災情報サービスプラットフォ－ムのデータ処理と中核機能

防災情報サービスプラットフォ－ム（以下、本プラットフォームと記す）はデータ処理における手順を定義した。

以下に示す。

図表 1‑22　本プラットフォームのデータ処理における手順



上記のようにデータ連携アダプタで接続し、データを取得する。そのデータをデータレイクへ蓄積する。蓄積されたデータを本プラットフォームの標準形式に変換し、データレイクへ保存する。保存されたデータを利用者のニーズ（サービス）に応じて複数のデータを重ね合わせ、集約、演算を行いデータの提供を行う。

上記の構成を自動に制御実行を行うアーキテクチャーとして処理の祖結合をデータドリブン・イベントドリブンで実施するイベントエンジン、データ処理を行う上でのメタデータやデータのFrom Toの関係を定義するデータディクショナリを定義、この2つのアーキテクチャーを合わせて自動処理アーキテクチャーとした。

自動処理のデータは、全てデータレイクを経由する。

本プラットフォームのデータ処理における手順で、データ層ではデータレイク、プラットフォーム層ではイベントエンジン、データディクショナリが本プラットフォームの中核機能に位置づけられると考えられた。

次章で各アーキテクチャーにおける方式設計を示し中核機能の説明とする。

# 中核機能個別の方式設計

本プラットフォームの基盤整備支援業務の仕様には、データレイク、イベントエンジン、データディクショナリそれぞれについて、定義すべき必要な個別の要件を記載している。

これらについても考慮し、設計すべき機能である。

## データレイクの方式設計

本プラットフォームの基盤整備支援業務の仕様には、本文中にデータ層に向けた仕様として多様なデータの種類や形式をサポートし、多様な操作を可能にする環境をつくること。セキュリティを担保し、データを連携させる範囲を設定できることとあり、更に以下の定義された要件がある。

1. 自治体等のユーザーレベルでデータ独立性を担保したセキュリティの確保。
2. 自治体等のユーザーレベルでデータごとに共有先の設定を可能にする。
3. 多様なデータ形式をそれぞれの特性に合わせた格納を可能にする。
4. データ形式や格納先を問わず、データの構造化・分析・集計のためのアクセスを可能にする。

中核機能の要件定義の内容をまとめて、上記のような要件に集約される。

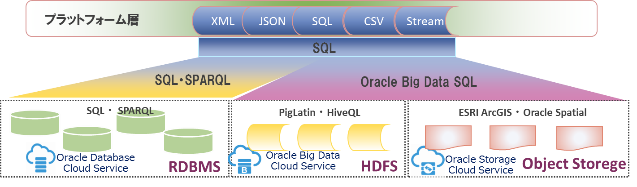
この内容を機能面の定義を行い、方式設計とする。

#### 多様なデータの種類や形式をサポートし、多様な操作を可能にする環境　及び多様なデータ形式をそれぞれの特性に合わせた格納を可能にする

本年度定義したプラットフォーム層の環境である、データインポート・提供環境より、多様なデータを連携することが可能となっており、データレイクについても多様なデータを保管・保存できる環境の提供が必要となる。

データレイクは、平成29年度より以下の構成としている。

図表 2‑1　データレイクの構成



上記のように、本プラットフォームのデータ保存環境であるデータレイクは、RDBMSをはじめ、HDFSやObject Storageの構成を取っている。

非構造データをはじめ、テキストや解析可能なバイナリーデータを含めて多様なデータの種類や形式をサポートしている。

図表 2‑2　データレイクの想定格納データ形式



平成29年度に検討された研究開発内容でも、以下の構造化されたデータの格納方式を定義している。

##### 構造データの格納形式

構造化されたデータは、データの同定により、形式や属性などの構造を保ち、基本的にRDBMS（テーブル）に格納する。

また、データの構造・形式によっては、RDBMS（テーブル）と元の文書の形式を保ち、並行して格納する。

###### テキストデータの格納形式

データレイク上の構造化されたデータのうち、テキストデータは文字コードをUTF-８とする。

また、日付のデータについては、日本時間を基本とする（GMT+9H）。

サービス層、データストアへの提供時に必要な形式への変換をサポートする。

テキストデータとしての格納は、文書の形式と構造化により、下記の格納形式とする。

###### 文字列型（構造化されたテキスト）

文字列型の構造化されたテキストデータは、属性ごとにテーブルの列として扱い、文字：VARCHAR2/CLOBでの格納、数値：NUMBER/ BINARY\_FLOAT・DOUBLEでの格納、日時：DATE/TIMESTAMPなどの形式にて格納する。

###### XML型テキスト

XML型は、災害対応データとしての必要な情報の範囲がテーブル構造化されていれば、上記の構造化（テーブル化）されたテキストと同様に扱うことを可能とする。

併せてXML型の元データは、XMLTypeデータ型として格納する。Oracleデータベースで用意されている、XML DB機能を利用しXMLデータの更新なども可能である。

###### JSONテキスト

JSONテキストは、災害対応データとしての必要な情報の範囲の属性が構造化されていれば、上記の構造化（テーブル化）されたテキストと同様に扱うことを可能とする。併せてJSONの元データは、CLOB/VARCHAR2データ型として格納する。

Oracleデータベースで用意されている、JSON関連の関数や検索処理を実施できる。

###### 非構造テキスト（テキストストリーム）

非構造テキストは、テキストストリームとしてCLOBに格納する。

この際、構造化の変換をした場合でUTF-８への変換が可能であった場合は、UTF-8で格納する。

変換にエラーが合った場合は、元のデータのまま格納する。

この場合BLOBに格納する可能性がある。

Oracleデータベースで用意されている、Oracle TEXT機能を利用して、自然言語処理を含むテキストマイニングの技術が利用可能である。

上記の構造化（テーブル化）されたテキスト、XMLテキスト、JSONテキストについて、相互に変換をする機能をデータ変換機能で実装するものとする。

###### 空間データ型（GISデータの格納形式）

空間データ型のひとつであるGISデータの扱いは、統計・分析に使用するため、標準的な形式としてOracle Spatial and Graphの形式であるWDSYS. SDO\_GEOMETRY型を基本とする。

逆GEOコーディングの為に、ArcGISの形式であるST\_Geometry型への変換をデータ変換機能で実装する。

同様に、プラットフォーム層での利活用にGEO\_JSON型への変換もデータ変換機能で実装し、相互変換を可能とする。

ShapeFile形式についても、SDO\_GEOMETRY型への変換もデータ変換機能で実装する。

###### 空間データ型（ラスターデータの格納形式）

空間データ型のひとつであるラスターデータについては、統計・分析に使用するため、標準的な形式としてOracleのネイティブな形式であるSDO\_GEORASTERデータ型を基本とし、メタデータやセルデータを透過的に格納する。

ESRI ArcGISでの扱いなどを含め、GEOTIFFへの変換などをサポートする。

上記のように、データレイクでの格納方式を定義している。

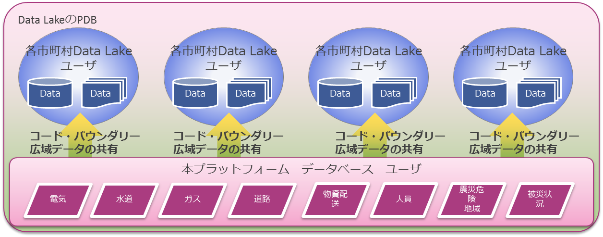
#### セキュリティを担保し、データを連携させる範囲を設定できる　及び　自治体等のユーザーレベルでデータ独立性を担保したセキュリティの確保

上記について、物理的に各自治体が使用するサービス及び領域を独立させる。

このために、基盤的にデータが独立する構成と考えている。

これは、平成29年度に検討された研究開発内容より継承している。

図表2‑3　データレイクの自治体が使用するサービス及び領域の独立概要

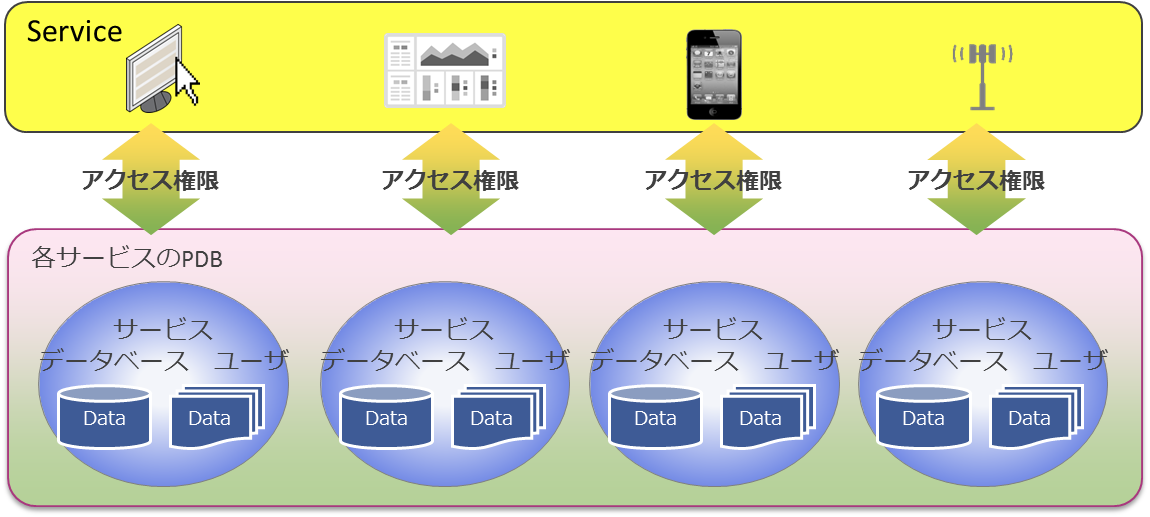


各サービスで利用されるデータベースインスタンスは、それぞれのサービス毎にデータベースのユーザーを用意する。

別のユーザー間のデータへアクセスはできない。

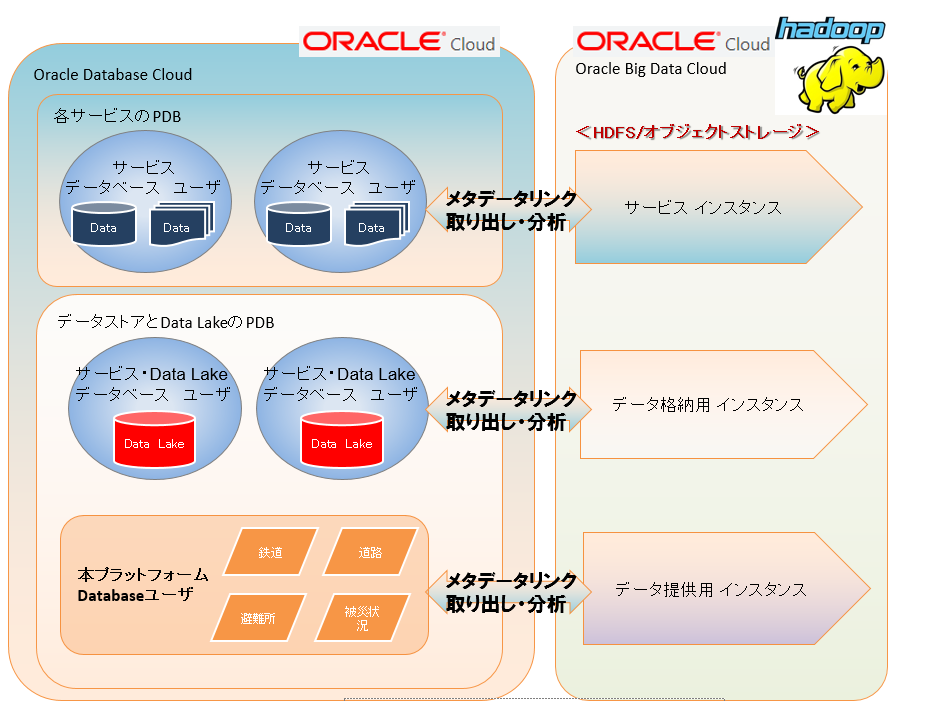
下図にて、各サービスで利用されるPDBのデータベースのユーザーを示す。

図表2‑4　データレイクのデータベースのユーザー概要



またデータレイクは、物理的に各自治体が使用するサービス及び領域をデータベースだけでなく、HDFSやオブジェクトストレージにも継承させる。

図表2‑5　データレイクのデータベースのユーザーとHDFS/オブジェクトストレージの独立性の関係



上記のように、ユーザーレベルでインスタンスや領域を分離する設計としている。

#### 自治体等のユーザーレベルで、データごとに共有先の設定を可能にする

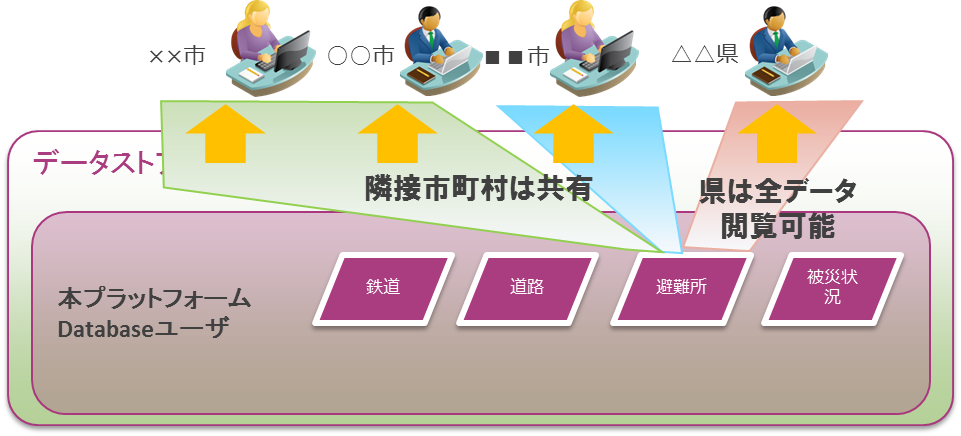
独立したセキュリティを確保するのと同時に、協定のある自治体間や、県への報告など他の自治体とのデータ共有が必要になる。

平成29年度に検討された研究開発内容では、以下と考えていた。

データベース機能のOracle Virtual Private Database及びデータベースのユーザー間（上記物理的な各自治体が使用するサービスの単位での個別のデータスキーマの共有を設定する）でデータ共有を考慮していた。

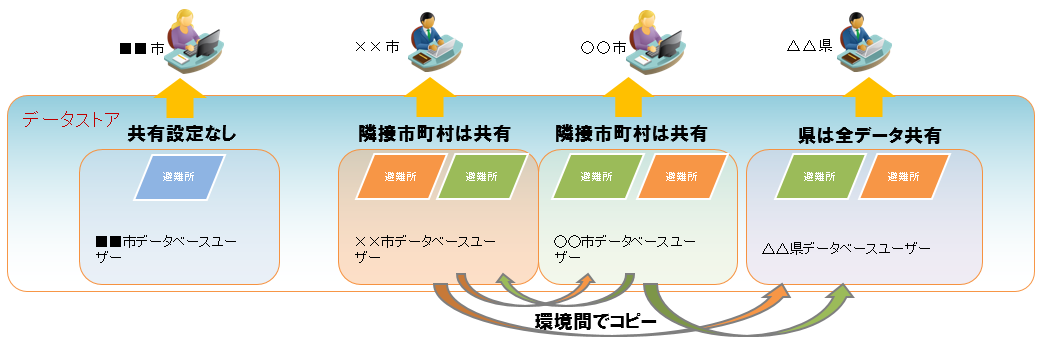
以下に、平成29年度に検討された研究開発内容を示す。

図表2‑6　29年度に検討された研究開発内容のデータ共有の概要



しかしながら複数のデータ・セキュリティ・ポリシーを適用すると、適用したポリシーの管理が複雑化しやすい傾向があり、管理の可視化にも影響するため、本年度は、データ連携処理として共有する自治体へデータをコピーすることでデータを共有することとする。

図表2‑7　本年度検討結果のデータ共有の概要



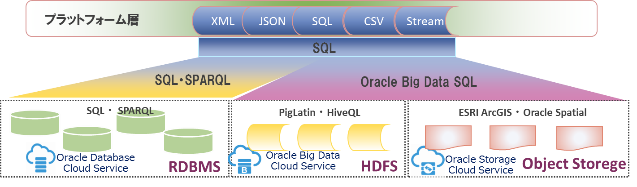
上記のような構成として、共有する際は「データをコピーするデータ処理を実施する」構成とした。

これにより、データのみ共有されている状態の不可視な状態をなくすことが出来る。但し、この構成の場合、コピーしたデータが元のデータと乖離するようなことが考えられる。そのためコピーしたデータが変更されないことを担保する為に分離して管理する。

#### データ形式や格納先を問わず、データの構造化・分析・集計のためのアクセスを可能にする

以下のように、それぞれの領域間は、DBの機能により連携・統合が可能であり、相互に分析が可能となっている。

図表2‑8　データレイクの構成

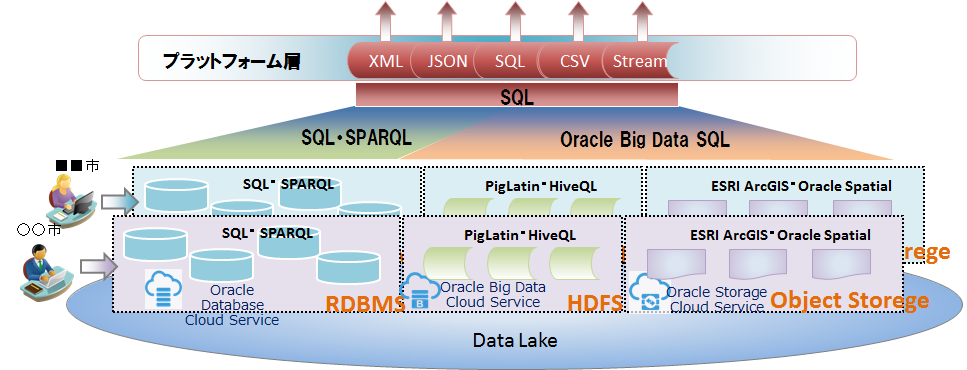


中核機能の要件定義にもあったDBMSを基本とし、容易に検索や部分抽出、分析、データ統合、レイヤ化を可能とするという要件にも対応できるように、オラクルクラウドのDBMSをはじめとするデータ格納環境は、SQLベースの操作でRDBMSをはじめ、HDFS/オブジェクトストレージの操作も一元的に可能となっている。

これと、セキュリティを担保する為のユーザーでの独立性を担保した構成とした。

以下に示す。

図表2‑9　一元的な操作と独立性を担保したデータレイクの構成



上記のように、全体の要件と個別に定義された要件を取り込んだ方式として、データレイクの方式とする。

上記と併せて、データレイク内での災害インシデントに応じた保存環境を提供するべきと考える為、以下の方式も定義設計する。

#### 災害インシデントの管理

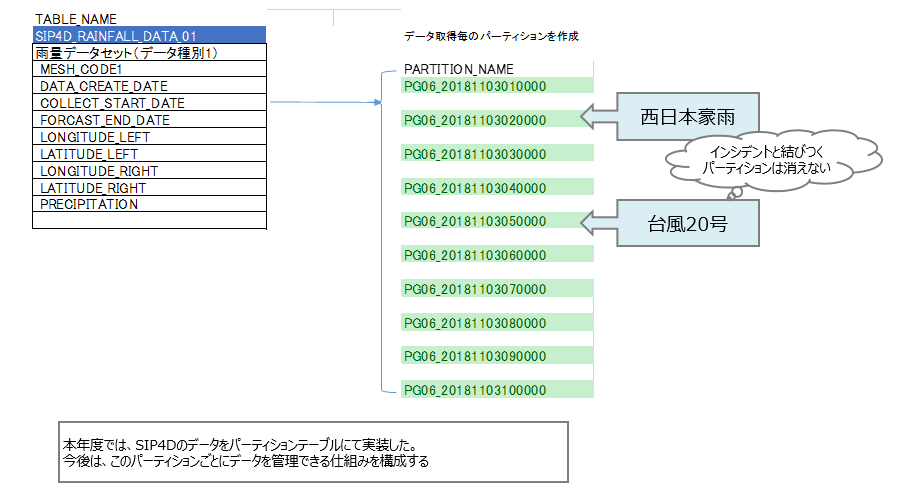
データレイクの保存データを、災害インシデントがあるデータと、それ以外のデータに分けて災害インシデントがある。データについてのデータの保存期間と、それ以外のデータの保存期間を分けられる構成とする。

この機能により、平時の利用で作成、利用したデータは、一定期間でデータを退避・圧縮しアーカイブ用に保存し、保有期間が過ぎたものから消えることになる。。

今期は、この構成をパーティションテーブルにて構成した。

以下に示す。

図表2‑10　パーティションテーブルの構成



上記は、雨量のデータをパーティション化した例である。

パーティション化によりパラレル処理を可能にし、処理速度の向上も可能となっている。

## イベントエンジンの方式設計

本プラットフォームの基盤整備支援業務の仕様には、以下の定義すべき要件の要素がある。

1. イベントとして、データの状態遷移を監視可能にすること
2. イベントとして、処理の状態遷移を監視可能にすること
3. イベントとして、時間の経過を監視可能にすること
4. エンジン機能として、上記3 つの状態遷移を同じように処理できること
5. エンジン機能として、処理負荷に連動し、処理リソースの変化を可能にすること
6. エンジン機能として、処理の経過や状況を管理でき、使用者に提供を可能にすること
7. 処理の定義をある程度簡素化できる仕組みを考慮できること

上記について、個別に要件と機能面の定義を行い方式設計とする。

また、以下の中核機能の要件定義で提示した要件についても、検討し構成を可能とする方式を設計する。

* 開発容易性

イベントエンジンの機能が、基本機能の範囲でも複雑で、最も品質を求められる部分であり、難易度が高いと考える。設計を十分に実施し、開発の範囲を極力なくす方向で検討したい。

* クラウドカバー機能

イベントエンジンの個々の機能については、できる限りクラウドで提供されるサービスを利用する。但し複数のエンジンを統合すること（それぞれがばらばらに動いても意味が無い）について検討する。

#### イベントとしてデータの状態遷移を監視可能にする

発災・状況変化に対応する為に、外部・内部での発生イベントをトリガーに、データ構造化処理を実行する機能が必要となる。

この機能は、Webサービス（SOAP）連携・REST・IoT連携における重要な機能となる。

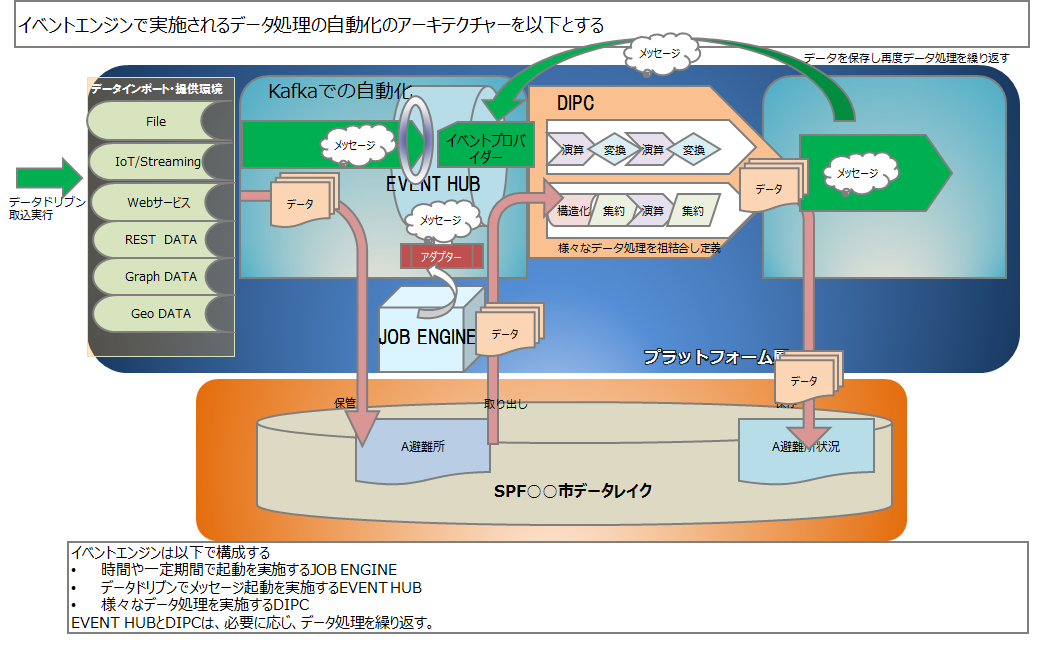
初期の想定は、Webサービス・RESTのメッセージをイベントとして、データ処理を実行するトリガーとして扱い、データドリブンとして自動処理を行う方式とする（データは、別として定義する）。

また、IoT等のストリーミングデータは、実際にストリーミングデータ処理機能として設計する。。

今期の構成では、オラクルクラウドで提供されるOracle Event Hub Cloud Serviceを利用し、Webサービス・RESTのメッセージをイベントとして、データ処理の自動化を行う方式として設計する。

以下にその構成を示す。

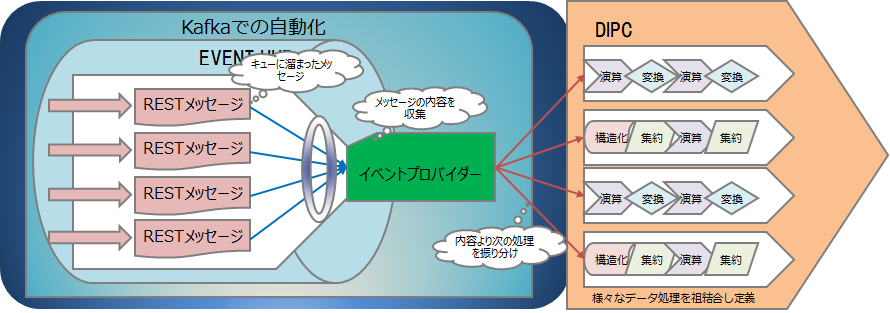
図表2‑11　 Event HUB機能を利用したメッセージをイベントとして扱い自動処理する方式



上記のように、Event HUBの中をWebサービス・RESTデータをメッセージとして扱い、次の処理への自動実行を促す。

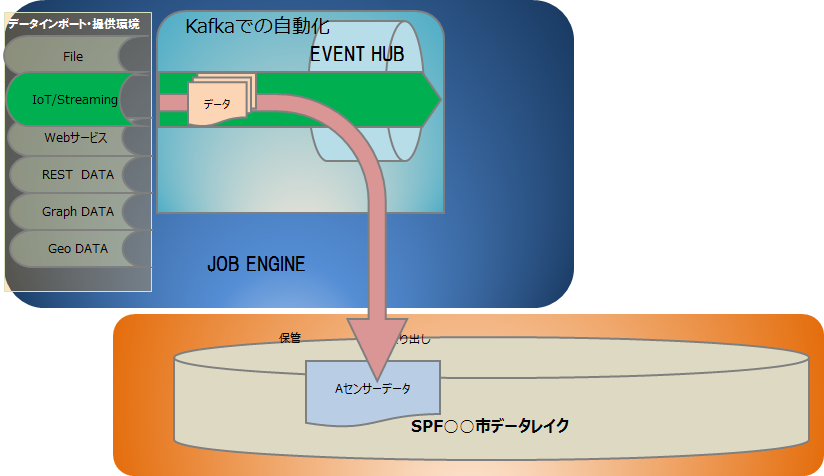
上記の構成には、Event Hubに蓄積したメッセージを順次取り出して、次の処理へ流すイベントプロバイダーが存在し、メッセージの内容により次の処理を実行する。

図表2‑12　 Event HUBのメッセージとイベントプロバイダーによる自動処理する方式



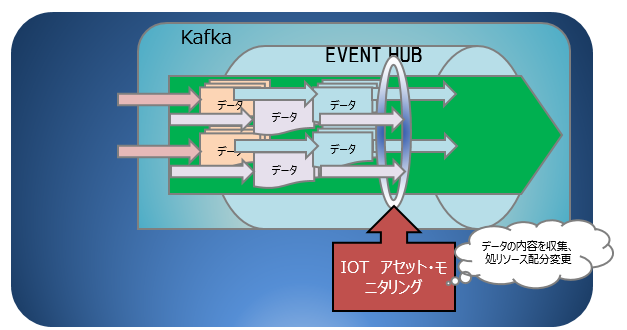
次に、IoTのストリーミングデータは、Event HUB機能を利用し、直接データ処理としてデータレイクへ蓄積する方式とする。

図表2‑12　 IoTのストリーミングデータのデータ連携方式



将来的には、IoTのストリーミングデータをデータの内容の偏り、データ内のキーワードを分析し、分析結果に基づく後続処理を実行させる。。

図表2‑14　 IoTのストリーミングデータの分析方式



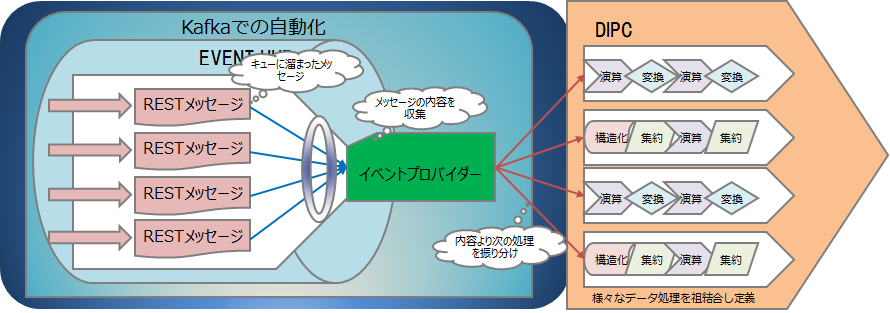
#### イベントとして処理の状態遷移を監視可能にする

データ処理を管理し、結果により処理や状態の遷移を行い、分岐しデータ処理を実行する機能が必要となる。

この機能は、本プラットフォームでのデータ処理を高度化し、より災害対応業務の自動化、災害対応業務の幅を広げる為の機能となる。

この機能は、上記のイベントプロバイダーより起動される本プラットフォームでのデータ処理を高度化において、オラクルクラウドで提供されるサービスの１つであるDIPC（Data Integration Platform Cloud）を構成し実施する。

図表2‑15　 Event HUBのイベントプロバイダーとDIPCによる自動処理する方式

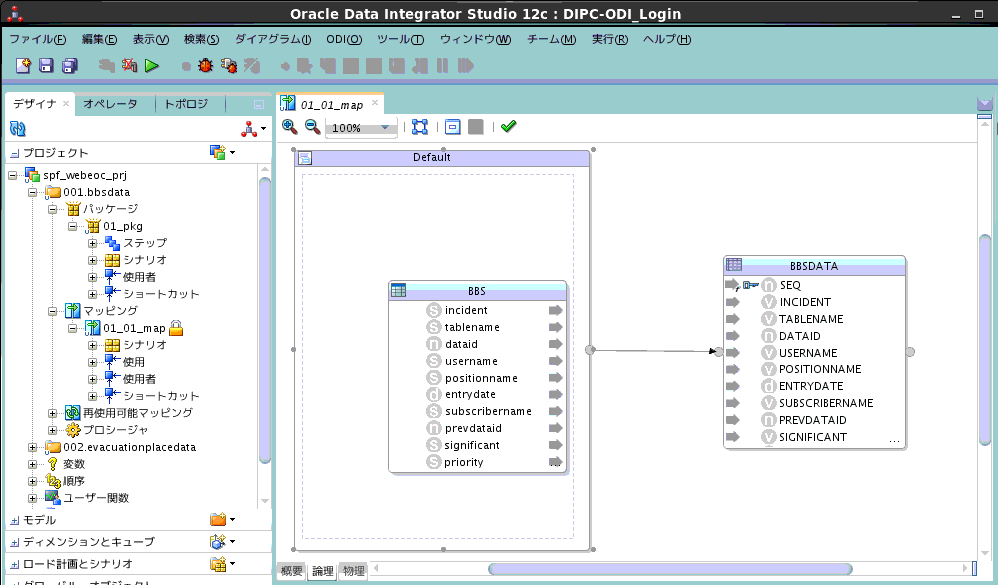
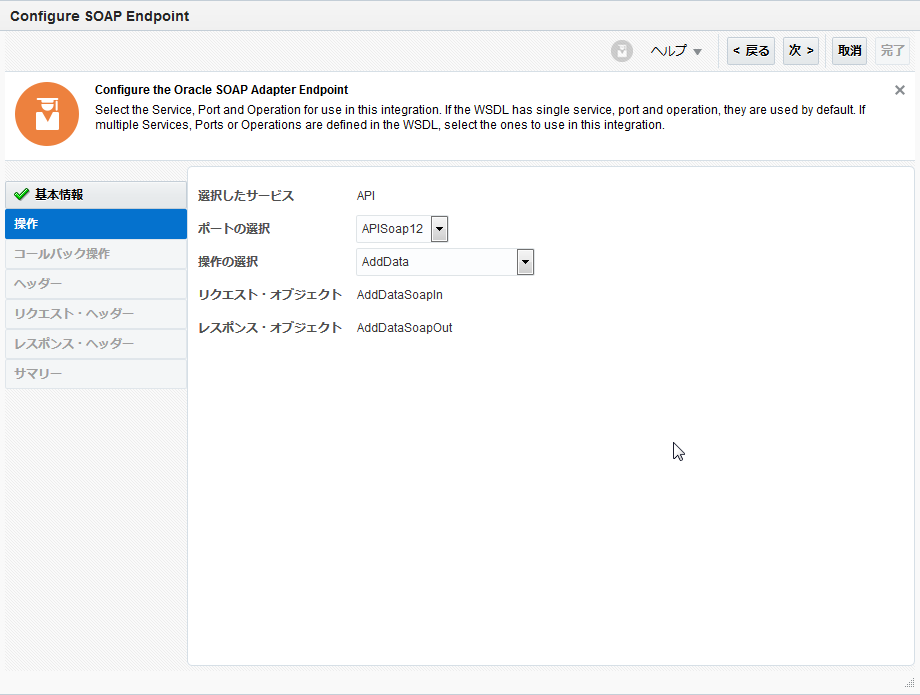


DIPCとは、データ・レプリケーション、統合、変換、ストリーミング、分析及びデータ品質管理を単一プラットフォームから実行できる機能であり、オラクルクラウド上のデータ統合機能の集合体となり、演算などを含むETLツールとなる。

DIPCは、上記のデータ処理をコンフィグベースで実装できる。

以下のような設定により、データ処理を構成できる。以下にその設定内容を示す。

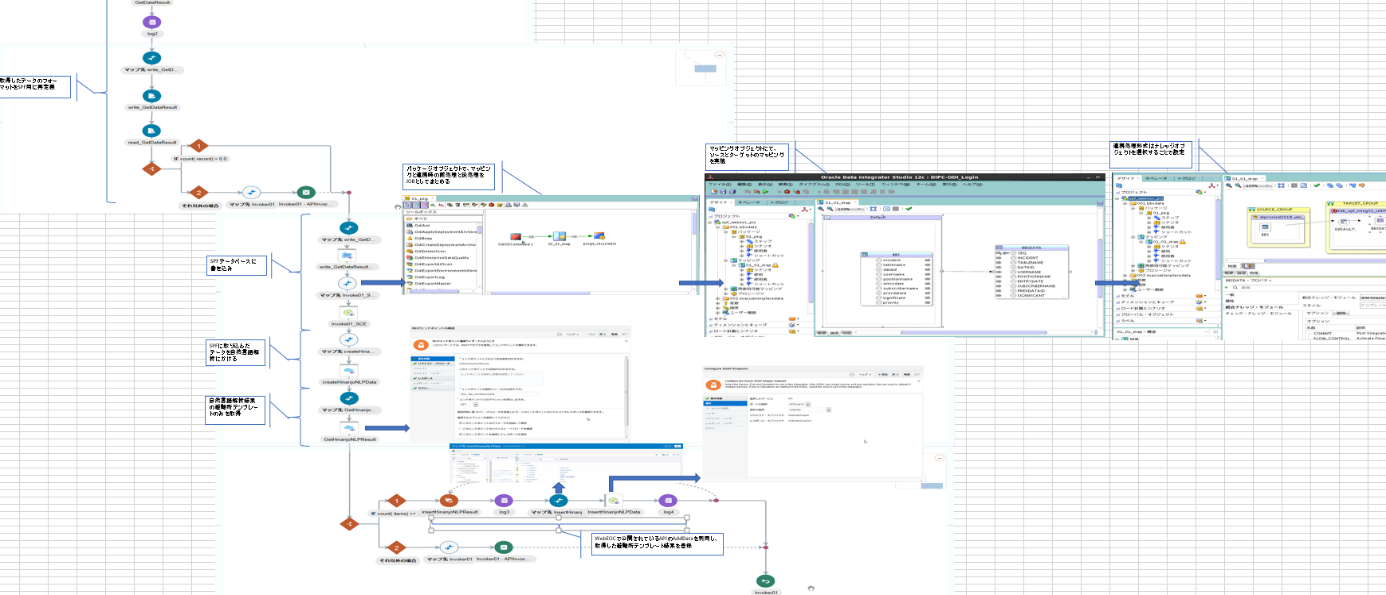
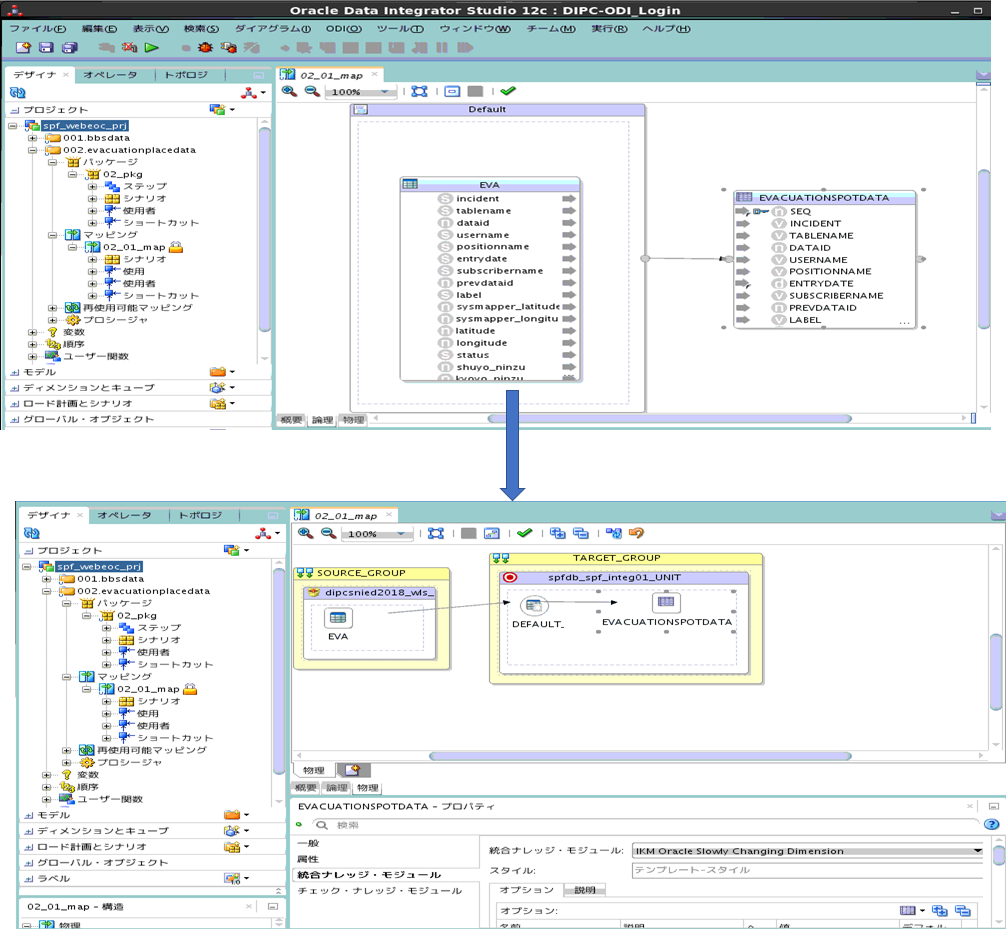
図表2‑16　 DIPCによるデータ処理の構成（接続構成）



上記は、データ処理を実施する為のデータ間の接続を定義している。

以下は、実際の処理の構成を設定している。

図表2‑17　 DIPCによるデータ処理の構成（処理構成）



上記のように、処理の分岐なども構成可能であり、DIPC内に基本的に用意されているデータ処理を構成でき、外部処理も用いることが可能となる。

#### イベントとして時間の経過を監視可能にする

イベントやデータドリブンとして連携できない場合、一定の経過時間をトリガーにデータ構造化処理を実行する機能が必要となる。

この機能は、外部サービスとの連携において必須な機能となる。

また、一定の経過時間のデータ収集により、時間経過による災害対応業務の状況遷移をデータ化することが可能になる。

但し、時系列や前の処理から一定時間経過後の再度処理するなどのサイクルを制御できる。

一般的に言うJOB ENGINE（JOB制御機能）となる、オラクルクラウドが現時点で提供されているサービスとして用意されていない。

このため、この部分は別の製品やOSSで賄う必要がある。

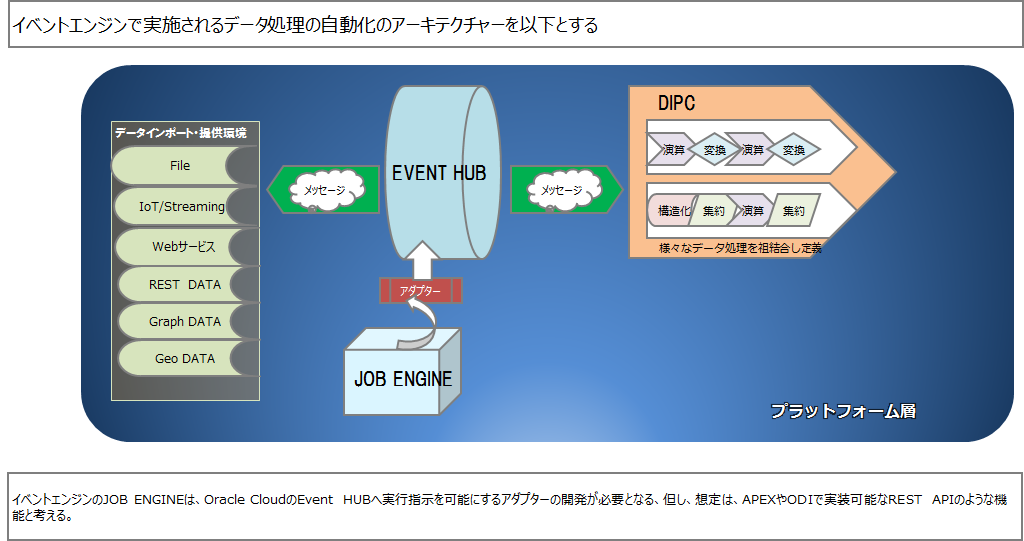
今回の要件は、実際の処理のトリガーは、一元管理することで管理が容易になることから、Event HUBのイベントプロバイダーより実行をする方法に統一したいと考える。

このため、JOB ENGINEは、時系列の管理と処理間隔の管理となり、HDFSでのJOB実行や、異OS間のJOB制御など、細かい機能は必要としない。

考慮が必要な部分は、JOB ENGINEからEvent HUBのイベントプロバイダーへのメッセージ連携を実施する、アダプタのような開発が必要となる点である。

この構成を以下に示す。

図表2‑18　 JOB ENGINEからEvent HUBのイベントプロバイダーへのメッセージ連携



このため、今期は製品の候補を選定するのみとし、今期の実装は、Cronもしくは、タスクスケジューラーでの実装とする。

上記の要件より、JOB ENGINEはOSSの製品として、以下の候補を挙げる。

* JobScheduler（<https://www.ossl.co.jp/?page_id=148>）
* hinemos （<http://www.hinemos.info/ja/hinemos>）

上記の製品で十分方式として成り立つとか考える。

#### エンジン機能として上記3 つの状態遷移を同じように処理できること

上記3つの機能を融合し、データ構造化処理を実行する必要があり、3つの機能を個々に構成するのではなく、1つのパッケージ機能として構成することが望ましい。

これにより、データ構造化処理を1つの機能で構成が可能となり、管理・構築面でもアドバンテージとなる。

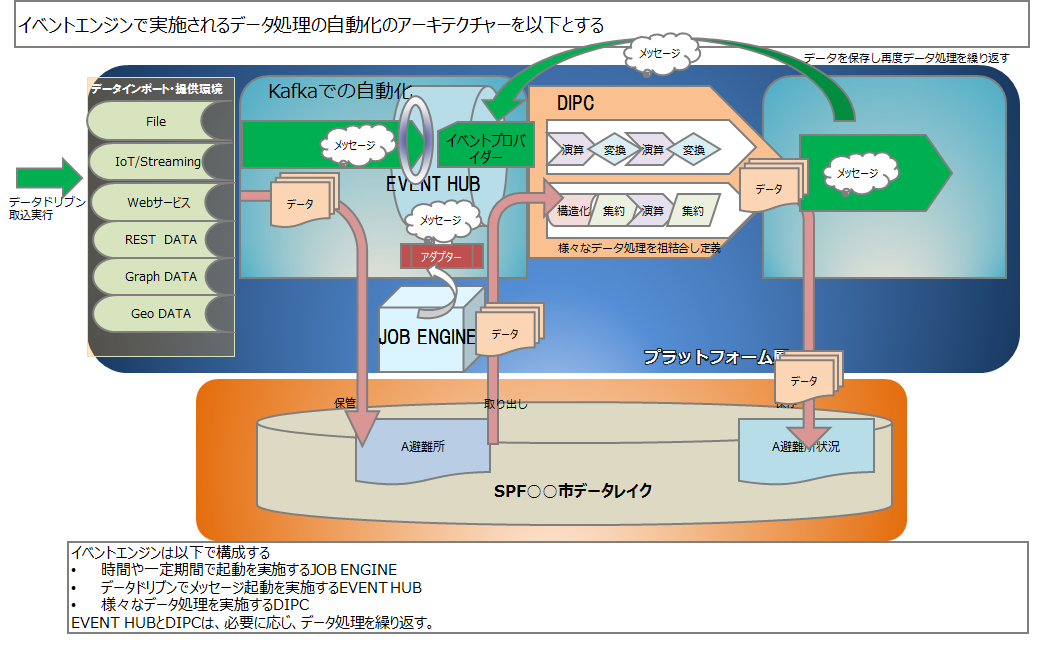
この構成による、本プラットフォームの全体的な運用コストへの寄与が大きいと考える。

今期構成した方式では、Event HUBのイベントプロバイダーを中心とした自動実行構成としている。

これと併せて、JOB ENGINEでの時系列での実行制御と、データ処理の実行遷移や分岐など、多様な内容をコンフィグベースで実装できるDIPCを構成し、再度データ処理が必要な場合は、更にメッセージでの連携を行い、疎結合する構成とする。

以下に示す。

図表2‑18　 今期構成したイベントエンジンの方式



上記の方式を取り、データ処理の自動化を実施する方式とした。

#### エンジン機能として処理負荷に連動し、処理リソースの変化を可能にする

この要件は、「(1) イベントとしてデータの状態遷移を監視可能にする」に記載した通りとなる。

初期としては、データ量やイベント数による処理のリソースを調整する機能を想定する。

将来的には、データの内容の偏り、データ内のキーワードを分析の結果を処理のリソースへ反映、機械学習へデータを提供する想定とする。

#### エンジンとして処理の経過や状況を管理でき、使用者に提供を可能にする

この要件は、本プラットフォームの管理運用として、また将来的には、利用者への提供機能として必要と考える。

災害対応業務は、一刻を争う状況が想定され処理の進捗や状況を、本プラットフォームの管理として把握する必要がある。

個々の処理単位の進捗は、見えない状況でも全体の進捗を管理できる必要がある。

この要件に対応するには、DIPCの大きな処理遷移時に、データベースへログを書き込むなどの方式の追加が必要となる。

#### 処理の定義をある程度簡素化できる仕組みを考慮できること

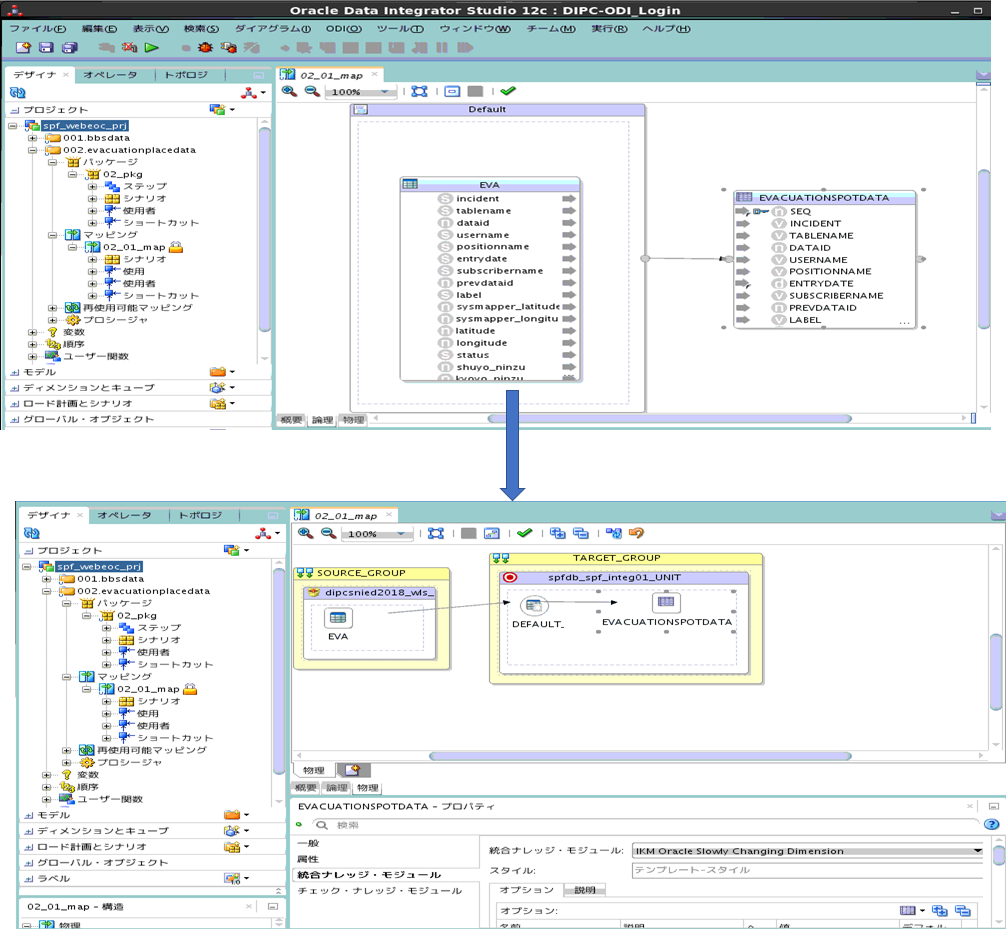
この機能は、「(4) エンジン機能として上記3 つの状態遷移を同じように処理できること」の記載と同じように、データ構造化処理の構成を簡素化するために必要となる。

災害対応業務の共通化（SOP）に沿って実行できるテンプレートや、過去に作成したデータ構造化処理をコピーできる機能などを想定する。

DIPCは、データ統合処理を連続して、様々なデータを取り込んで処理が可能である。

以下に示す。

図表2‑20　 DIPCのデザイン機能



このように、データ処理として、SOPや業務を模した構成とすることで一般化することを想定する。

また、防災情報としての語彙の増やし、語彙に関連するデータ処理をテンプレート化することで、一般化が可能かと思われる。

## データディクショナリの方式設計

本プラットフォームの基盤整備支援業務の仕様には、以下の定義すべき要件の要素がある。

1. ディクショナリとして、データの形式を問わず、メタデータ（形式や属性情報）を登録可能にすること
2. ディクショナリとして、データ種類毎のデータカタログへ登録する情報を登録可能にすること
3. ディクショナリとして、データ種類毎の共有範囲を設定可能にすること
4. ディクショナリとして、データ毎の管理機能（保管期間等）を提供できること
5. 防災データ連携の標準化・高度化を目的に、防災分野の語彙を災害対応業務の流れと併せて定義すること
6. 上記で作成する防災分野の語彙は、共通語彙基盤と整合性をとること
7. ディクショナリの定義をある程度簡素化できる仕組みを考慮できること

上記について、個別に用件と機能面の定義を行い、方式設計とする。

また、以下の中核機能の要件定義で提示した要件についても検討し、構成を可能とする方式を設計する。

* リポジトリの考え方（クラウドアーキテクチャーの採用範囲）によって大きく変わる、EEやデータ連携処理にデータを、必ず提供できる機能を有する必要がある。

#### ディクショナリとしてデータの形式を問わず、メタデータ（形式や属性情報）を登録可能にする

この要件を設計するに当たり、データの形式や、属性を想定する必要があり、この部分は前述のデータレイクへ格納できるデータ形式や、格納形式を網羅するものとする。

概要は以下となる。

図表 2‑21　データレイクの想定格納データ形式



##### 構造データの格納形式は、以下を想定している。

###### テキストデータ

###### 文字列型（構造化されたテキスト）

###### XML型テキスト

###### JSONテキスト

###### 非構造テキスト（テキストストリーム）

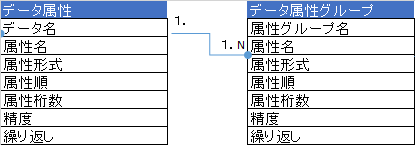
###### 空間データ型（GISデータの格納形式）

###### 空間データ型（ラスターデータの格納形式）

上記のうち、(ウ)XML型テキスト、(エ)JSONテキスト、(カ)空間データ型（GISデータの格納形式）、(キ)空間データ型（ラスターデータの格納形式）は、定義された繰り返しを含む複数の属性から構成されており、この部分を考慮した設計が必要となる。

以下に、今期設計したメタデータ（形式や属性情報）の格納モデルを示す。

図表 2‑22　メタデータ（形式や属性情報）の格納モデル



上記のデータ属性は、一般的な属性を定義する。

これに、XMLやJSONなどで定義される、ネストした属性を定義管理するデータ属性グループを定義した。

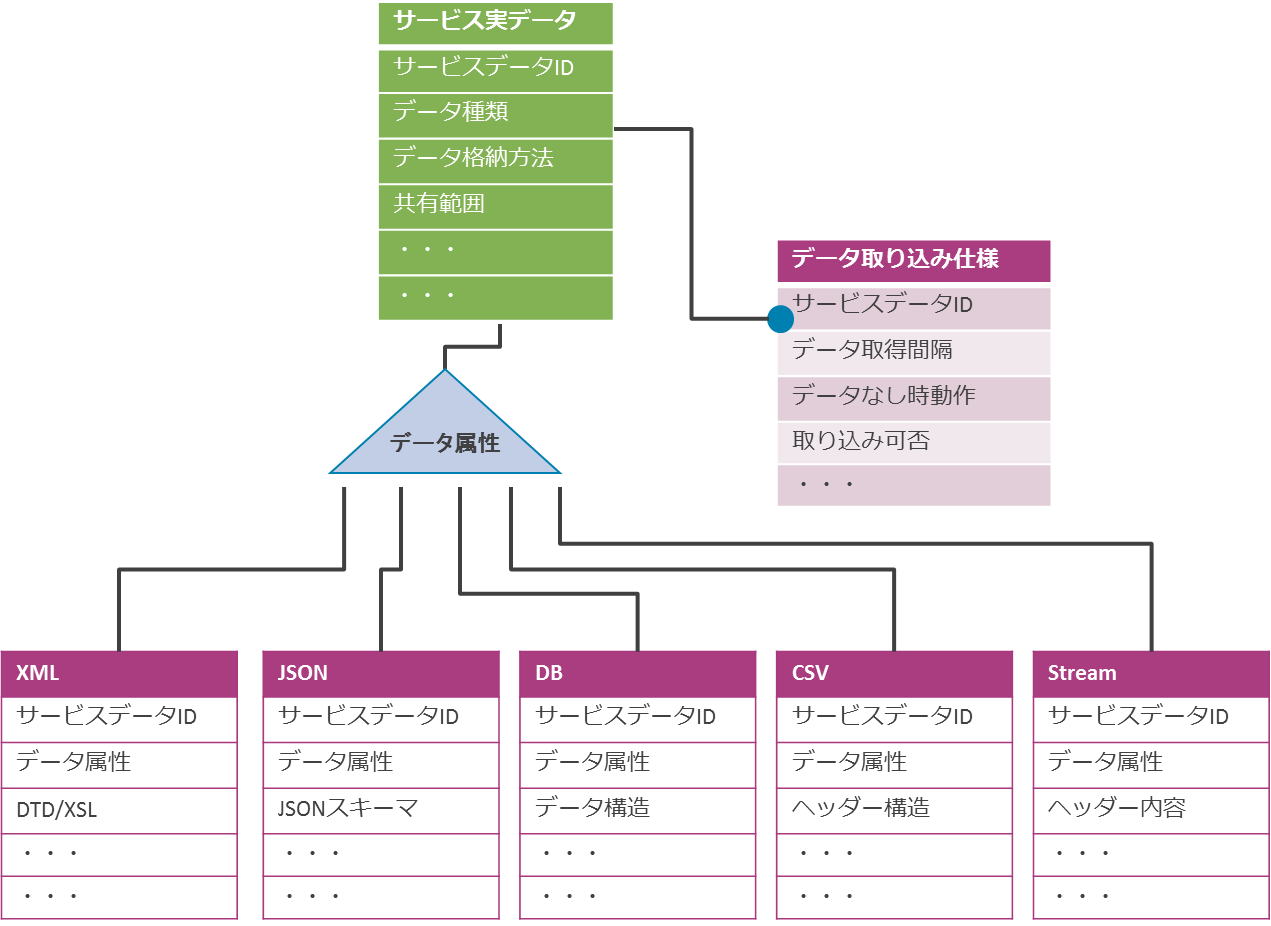
この内容で、ネストした構造の属性グループを定義し、さらにネスとするような構成の場合は、データ属性グループの属性名に、属性グループ名が入る自己従属を構成する。

上記のような設計とした。

#### ディクショナリとしてデータ種類毎のデータカタログへ登録する情報を登録可能にする

平成29年度に検討された研究開発内容では、データカタログをCKANとして想定した設計を行った。

図表 2‑23　平成29年度に検討されたデータカタログ



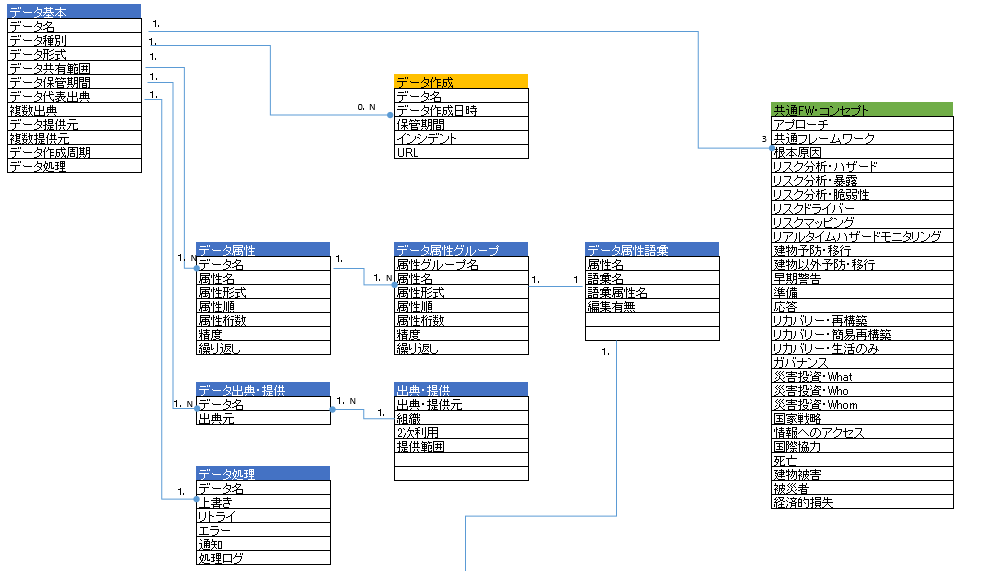
上記のサービス、実データに、CKAN上でのメタデータを登録する構成を考えていた。

今期は、CKANだけではない多様なデータカタログに対応できるような構成を検討した。

また、今期は、別の役務でPORTAL機能を実装しており、PORTALとの連携も意識した構成とした。

以下の構成とした。

図表 2‑23　データカタログのメタデータ収録部分



上記のような構成とし、データ基本でメタデータの基本情報（CKANのメタデータなども含む）を登録し、データ出典・提供で出典や、2次利用可能な情報などの管理を可能にした。

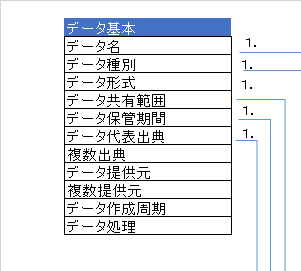
また、動的なデータ登録を可能にするためデータ作成で、イベントごとにデータ処理され、提供可能になるデータを管理できる構成としている。

併せて、共通FWコンセプトにてPORTAL上での検索タグを提供できる構成とし、今期のデータディクショナリの設計とした。

#### ディクショナリとしてデータ種類毎の共有範囲を設定可能にする

共有設定については、ユーザー管理と同期を取る必要があるが、今期はこの範囲にないため、この部分を想定した構成を設計した。

図表 2‑25　データカタログの共有設定部分



上記の通り、データの基本情報として、データ共有範囲として構成した。

共有範囲の内容は、ユーザー管理で行われるグループ定義を想定している。

この部分がネストする構成や複雑化する場合、この属性を分離した構成を定義する必要がある。

#### ディクショナリとしてデータ毎の管理機能（保管期間等）を提供できること

共有範囲と同様に、データの基本情報として保管期間を定義できるものと定義した。

但し、基本的には災害インシデントのデータに関しては、永年保管する想定としている。

それ以外のデータは、この保管期間を基本としたデータライフサイクルとする。

この基本期間が設定されていないデータについては、2ヶ月（60日）を保管期間とする。

#### ディクショナリの定義をある程度簡素化できる仕組みを考慮できること

ディクショナリを、災害対応業務の共通化（SOP）に沿って実行できるテンプレートや、過去に作成したディクショナリをコピーできる機能などを想定する。

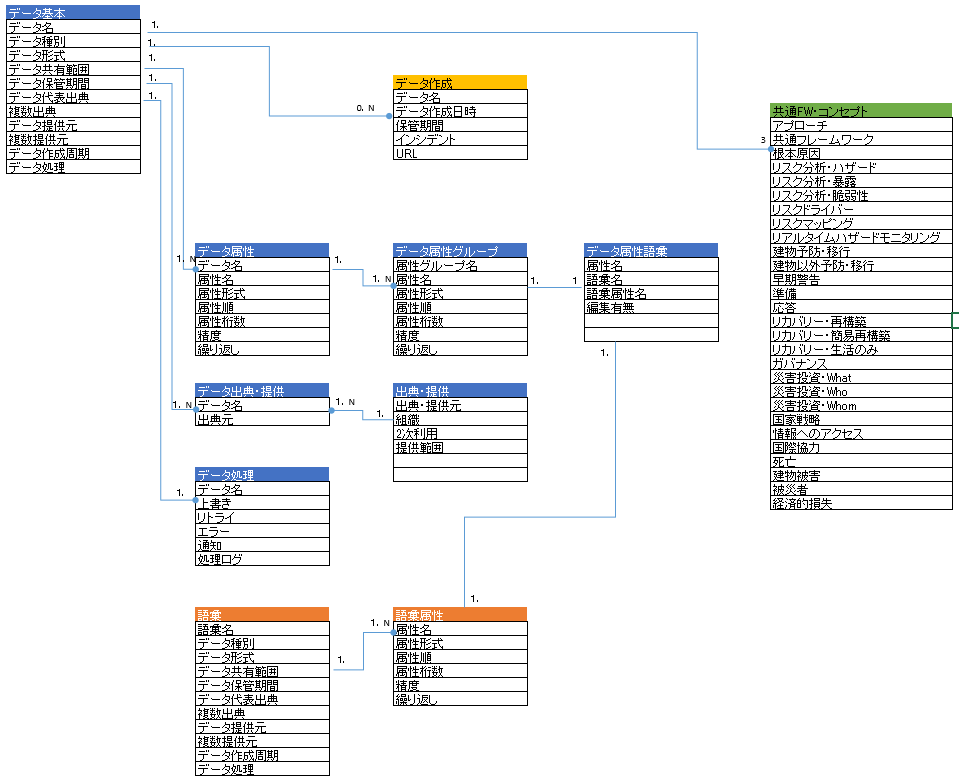
また、上記の災害対応業務の語彙を拡充し、これもテンプレート化する構想とする。

このためにも、データレイクへデータを多く登録できるように進める。

#### 今年度設計のデータディクショナリのデータモデル

以下に、今年度設計したデータディクショナリのデータモデルを示す

図表 2‑26　今年度設計したデータディクショナリのデータモデル



## データ処理の設計

データ処理の設計及び実装については、今期実装した内容に基づき記載する

### 今期実装のデータ処理

今期実装したデータ処理は、以下となった。

##### 雨量データのGEOデータ化

##### 地震データのGEOデータ化

##### WebEOC（橿原市訓練データ）の連絡票テキストデータの自然言語処理

上記の実装を実施した。

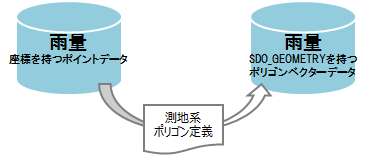
### 今期実装のデータ処理の設計

#### 雨量データのGEOデータ化

雨量データは、SIP4DよりFTPでデータを取得し、データをRDBへロードする。

このデータを、GEOデータ（ベクター）型のデータレイクへの標準格納形式であるOracle Spacial and Graphの形式である、SDO\_GEOMETRY型へ変換し、格納するものである。

図表 2‑27　雨量データのデータ処理



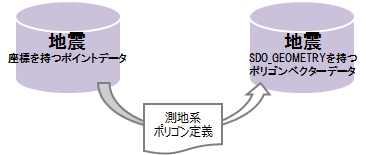
上記の処理を、SIP4Dとのデータ連携及びデータ処理に加え実装を行う。

#### 地震データのGEOデータ化

地震データは、SIP4DよりFTPでデータを取得し、データをRDBへロードする

このデータをGEOデータ（ベクター）型のデータレイクへの標準格納形式であるOracle Spacial and Graphの形式である、SDO\_GEOMETRY型へ変換し格納するものである。

図表 2‑28　地震データのデータ処理



上記の処理をSIP4Dとのデータ連携及びデータ処理に加え、実装を行う。

#### WebEOC（橿原市訓練データ）の連絡票テキストデータの自然言語処理

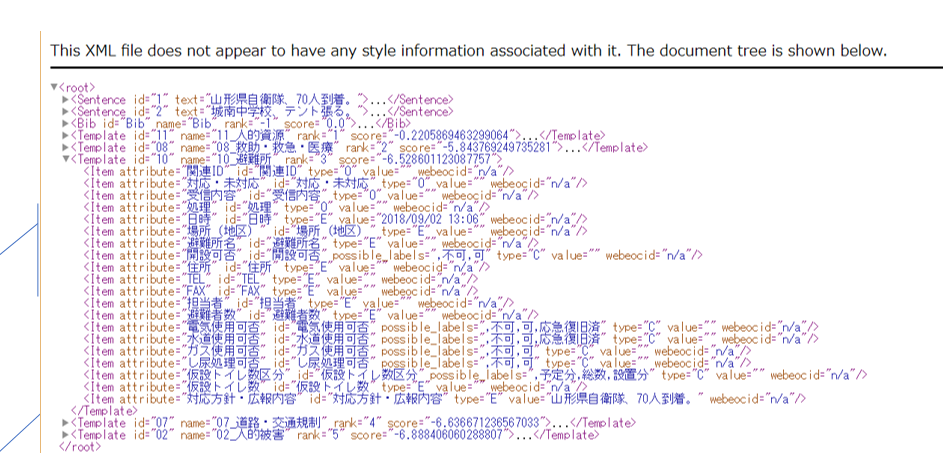
WebEOC（橿原市訓練データ）の連絡票の非構造テキストデータをデータ連携し、このテキストの内容を自然言語処理し、自然言語処理の避難所テンプレートに当てはまった内容を、構造化したXML（SOAP）でWebEOCへ再度返却する処理を実装した。

この内容の中でのデータ処理は、連絡票テキストデータを自然言語処理し、自然言語処理の避難所テンプレートに変換する部分となる。

本年度実施した自然言語処理は、RISTEXー防災情報データベース化支援システムであり、これを現在OracleCloudのIaaS上にデプロイしており、本プラットフォーム上より呼び出しが可能な状態とした。

以下に自然言語処理解析結果を示す。

図表 2‑29　自然言語処理解析結果



上記のように解析される。

上記のようにTemplateごとに解析結果を返す仕様となっており、上記の避難所の部分をWebEOCへ返す仕様とした。

以上の内容を中核機能の方式設計として定義する。