

# Bluemixを活用したIoT基盤構成サンプル



Bluemixはいろいろな目的のシステム基盤として利用できますが、その一例としてIoT PoC環境を実装することも可能です。

以下にIoT PoC環境を構成する上で標準的に必要な構成要素を列挙します。

Bluemixではこれらのサービスを立ち上げ、初期設定を行うだけで簡単にIoT PoC環境を構築することが可能です。

(1) IoT Platform


(2) Node-RED




(3) Cloudant

(4) dashDB

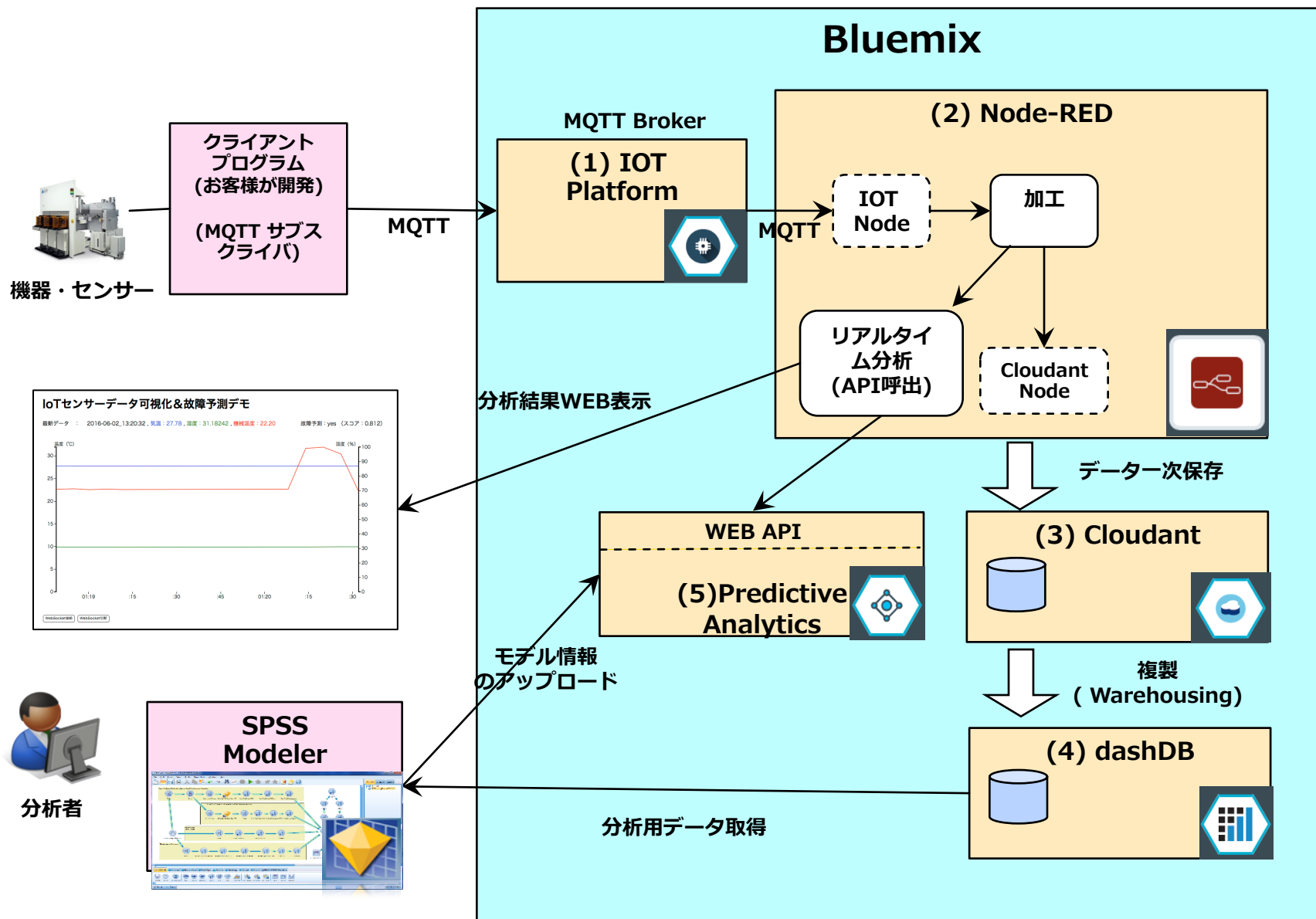
(参考) SPSS Modeler (PC側のソフト)

(5) Predictive Analytics

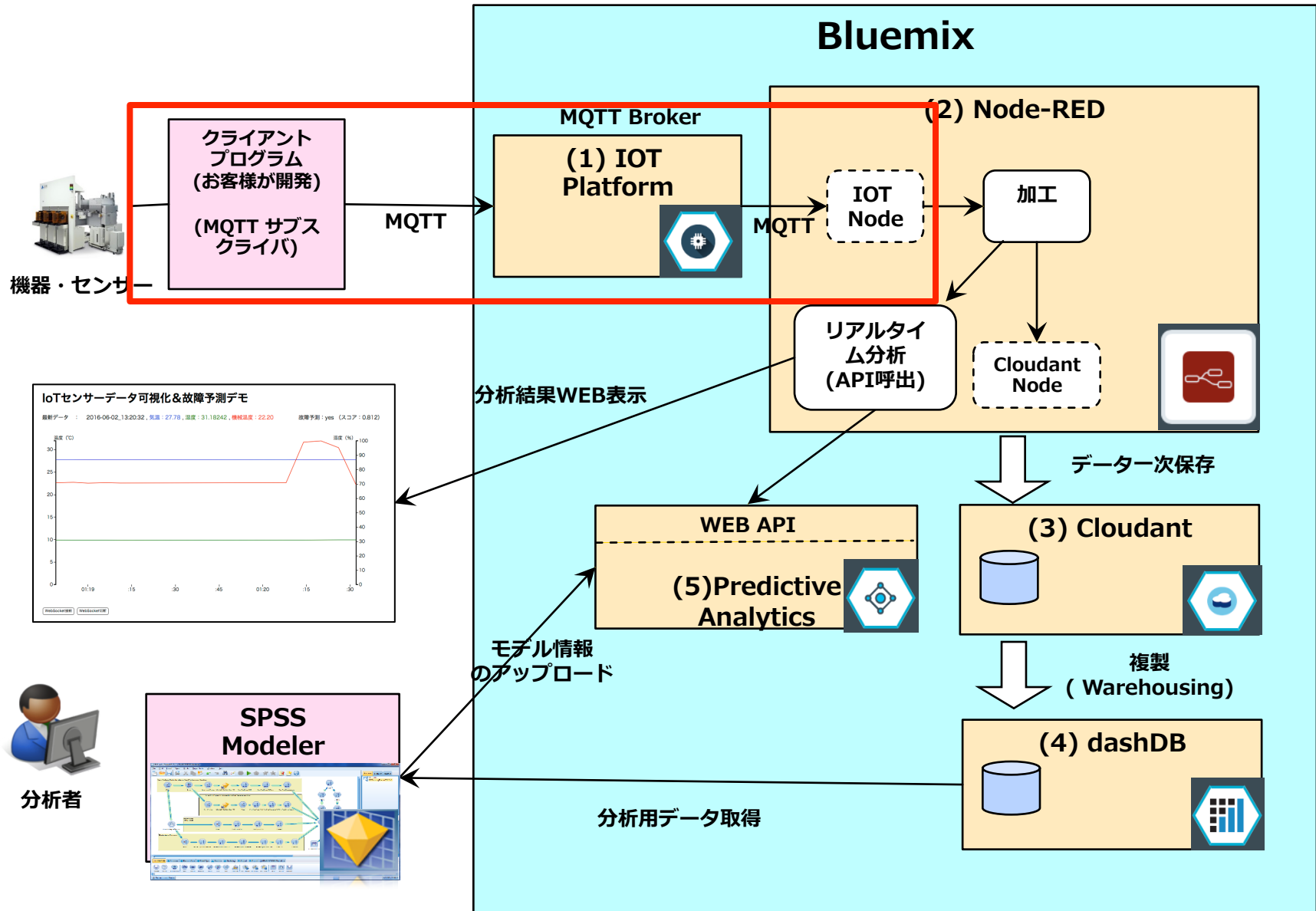
| アプリケーション  |   |                  |  |
|---|---|------------------|--|
| アプリ名  |   | ランタイム            | 正常性                                      |
|  aka-nodered | <a href="http://aka-nodered.mybluemix.net">http://aka-nodered.mybluemix.net</a> | SDK for Node.js™ | <span style="color: green;">●</span> 実行中 |

| サービス   |                             |
|--|-----------------------------|
| サービス名  | サービス・オフリング                  |
|  cloudant-NoSQL-DB     | Cloudant NoSQL DB           |
|  dash-DB               | dashDB                      |
|  IoT Platform          | Internet of Things Platform |
|  Predictive Analytics | Predictive Analytics        |

次ページ以降でそれぞれの構成要素の役割と特徴について説明します。



# (1) IOT Platform (Internet of Things Platform)





## IOT Platform の役割

MQTT(注)を利用したIoTデータ収集のためのフロントサーバーです。

## 特徴

- Publisher – Subscriber モデル採用による疎結合なデータ連携の仕組みで1対多、多対多の通信を容易に実現可能です。
- IOTデータ収集のために考えられた軽量プロトコルで大量データ受信に最適です。
- データ重要度とシステム負荷のトレードオフを考慮して3種類のモードを選択可能です。  
(「最高1回」「少なくとも1回」「正確に1回」の3種類)
- セキュリティ確保のため、SSL通信も可能です。

(注)

MQTTは1999年にIBM社とEurotech社によりIOTデータ収集を目的として考案されたプロトコルです。

現在OASISという国際標準化団体で標準化を検討中です。

## ▪ Publisher – Subscriber モデル

データの送り手(Publisher)と受け手(Subscriber)の間に必ず仲介者(Broker)が入るモデルです。

Publisher とSubscriberはそれぞれBrokerとの通信のみ気にすればよく、最終的な通信相手のことは一切意識しなくていいことがその特徴となります。

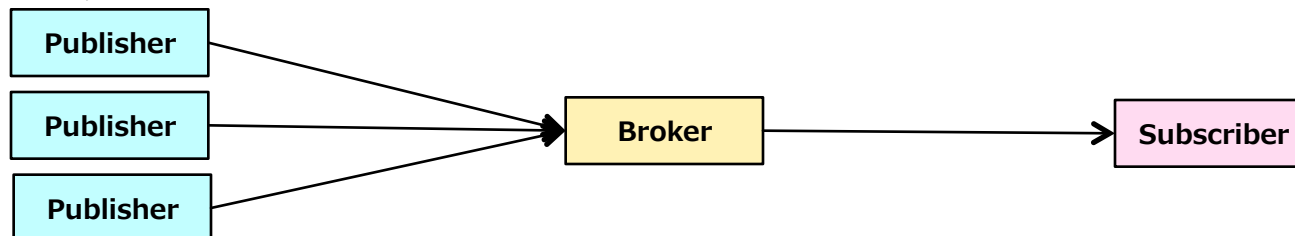
-> システム間連携が疎となり、大規模システムを容易に構築可能となります。

(例) 下記のどのパターンもアプリ側の修正なしに実現可能です。

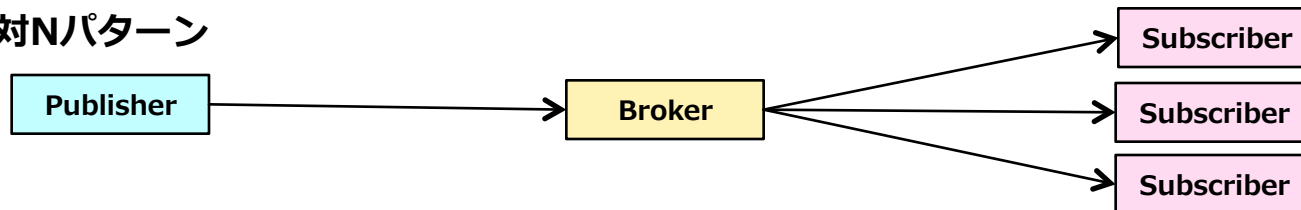
### 基本パターン



### N対1パターン

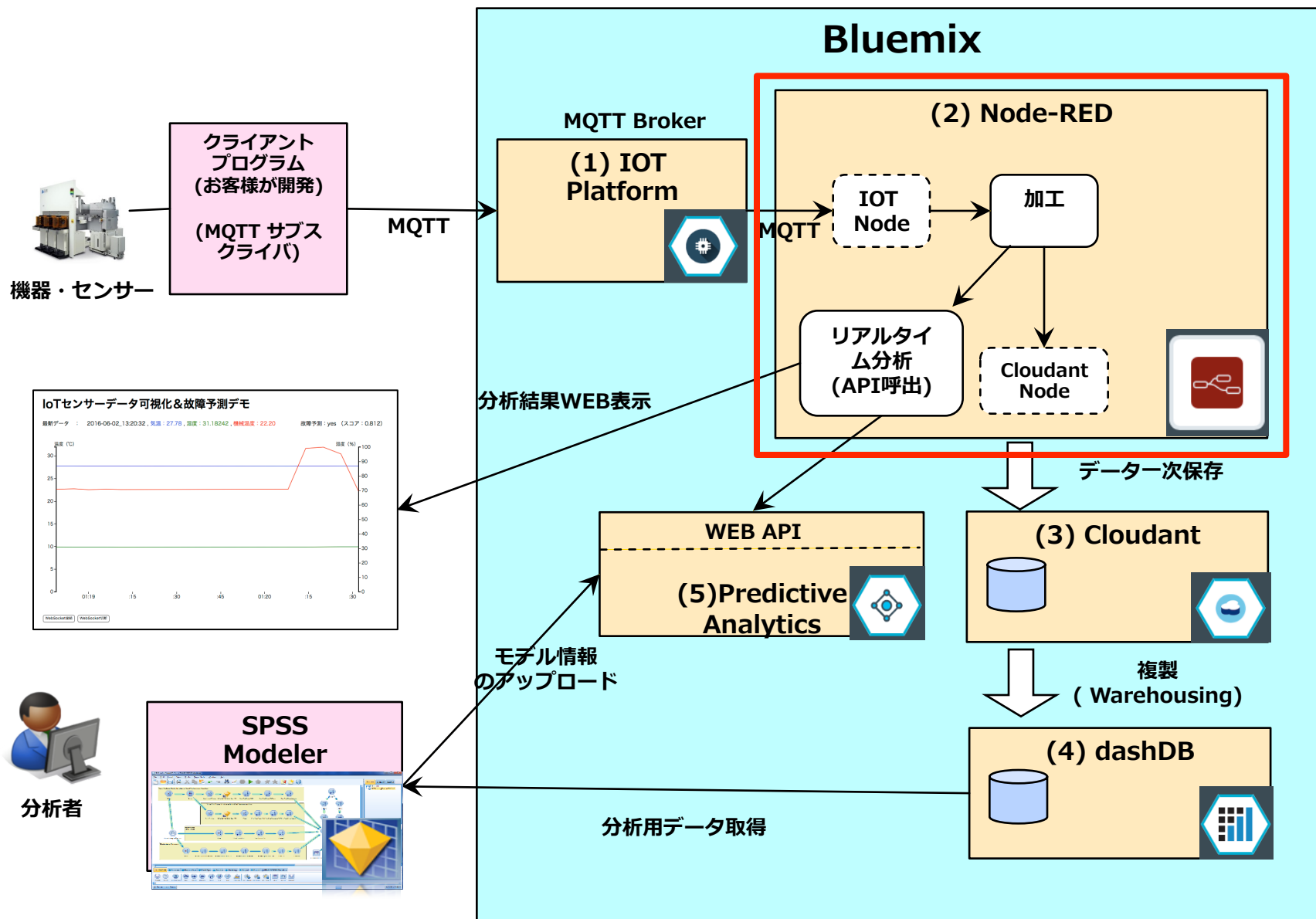


### 1対Nパターン



(ブランクページ)

## (2) Node-RED







### Node-RED の役割

Bluemix上でサービス間の連携を担うコントローラです。

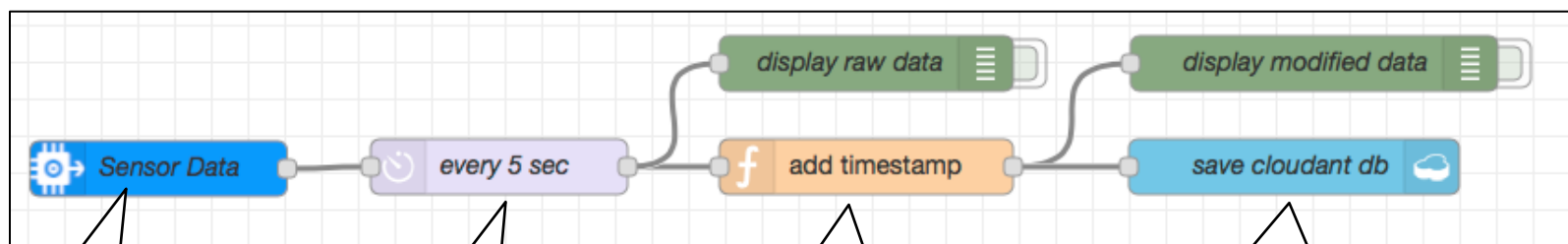
### 特徴

Bluemix上の他サービス、Bluemix以外の外部APIサービスをNodeで表現し、Node間の連携をGUI上の接続により簡単に実現できます。

細かいロジック処理はJavaScript処理ノードにより、コーディングで対応可能です。

上記特徴を活用することで、PoCアプリの超高速開発を実現します。

### Node-REDのフロー例



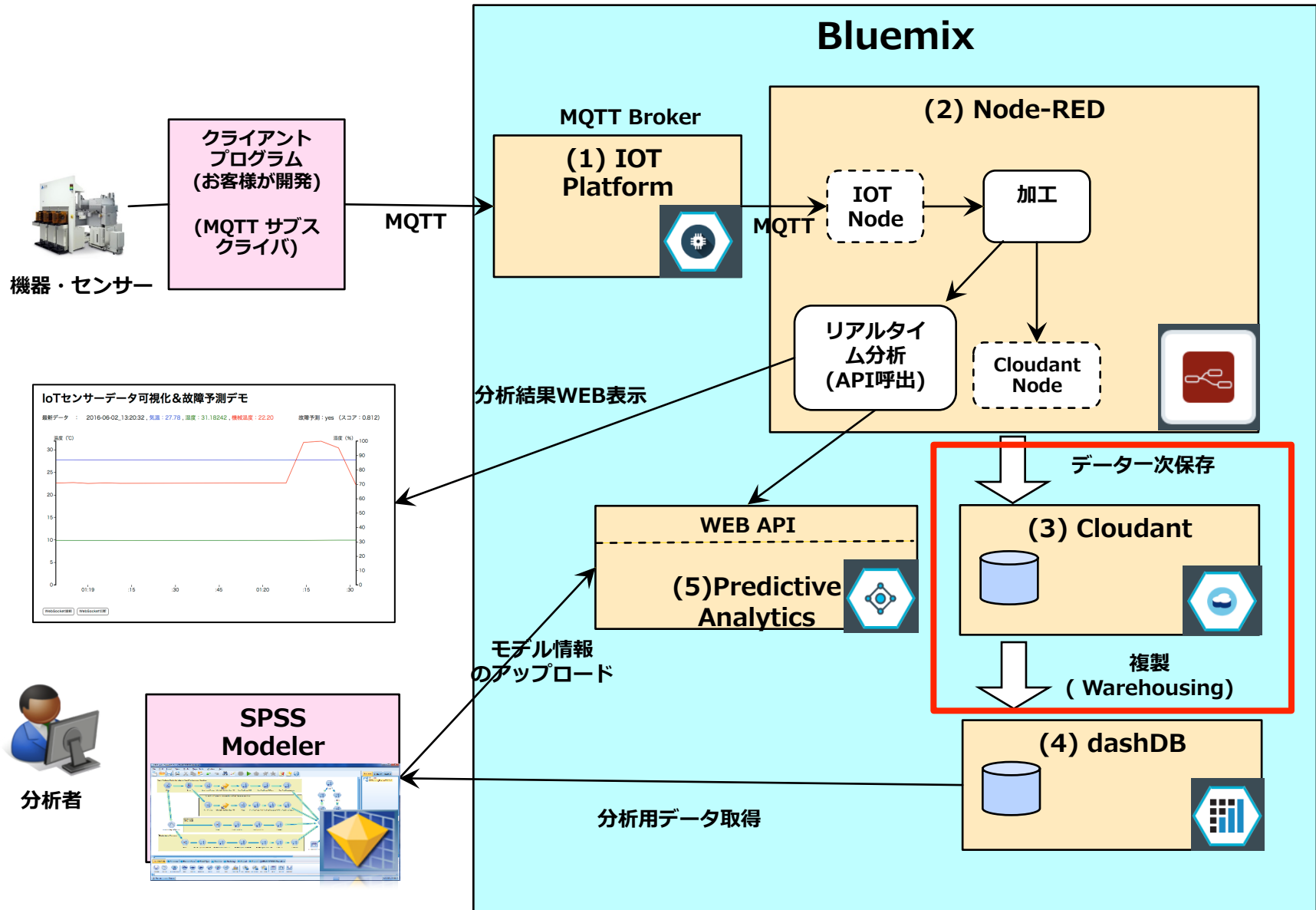
センサー入力データノード  
(MQTTサブスクライバ)

毎秒到着するデータを  
5秒に1回に間引く

JavaScript処理ノード  
(センサーデータにタイム  
スタンプを付加)

Cloudantノード  
(センサーデータをCloudantに保存)

# (3) Cloudant



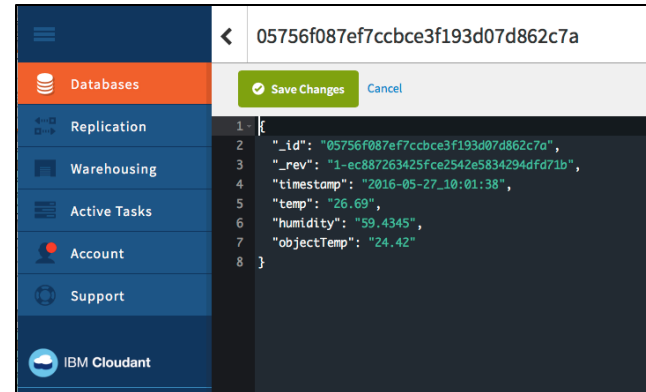
## Cloudbant の役割

IoTデータの一次保管先DBです。

## 特徴

- ・ **NoSQL型データベース**(ドキュメント指向型)

RDBで必要なデータ項目設計なしに、JSON型データで任意の項目を保存可能です。



- ・ **運用・非機能要件への対応**

DBの運用はクラウド側で実施するので、ユーザーは意識する必要がありません。(マネージドサービス)  
サーバー構成はデフォルトで3台構成(マスターマスタ型構成)です。

データも冗長化されており、ユーザー側はバックアップを考慮する必要がありません。

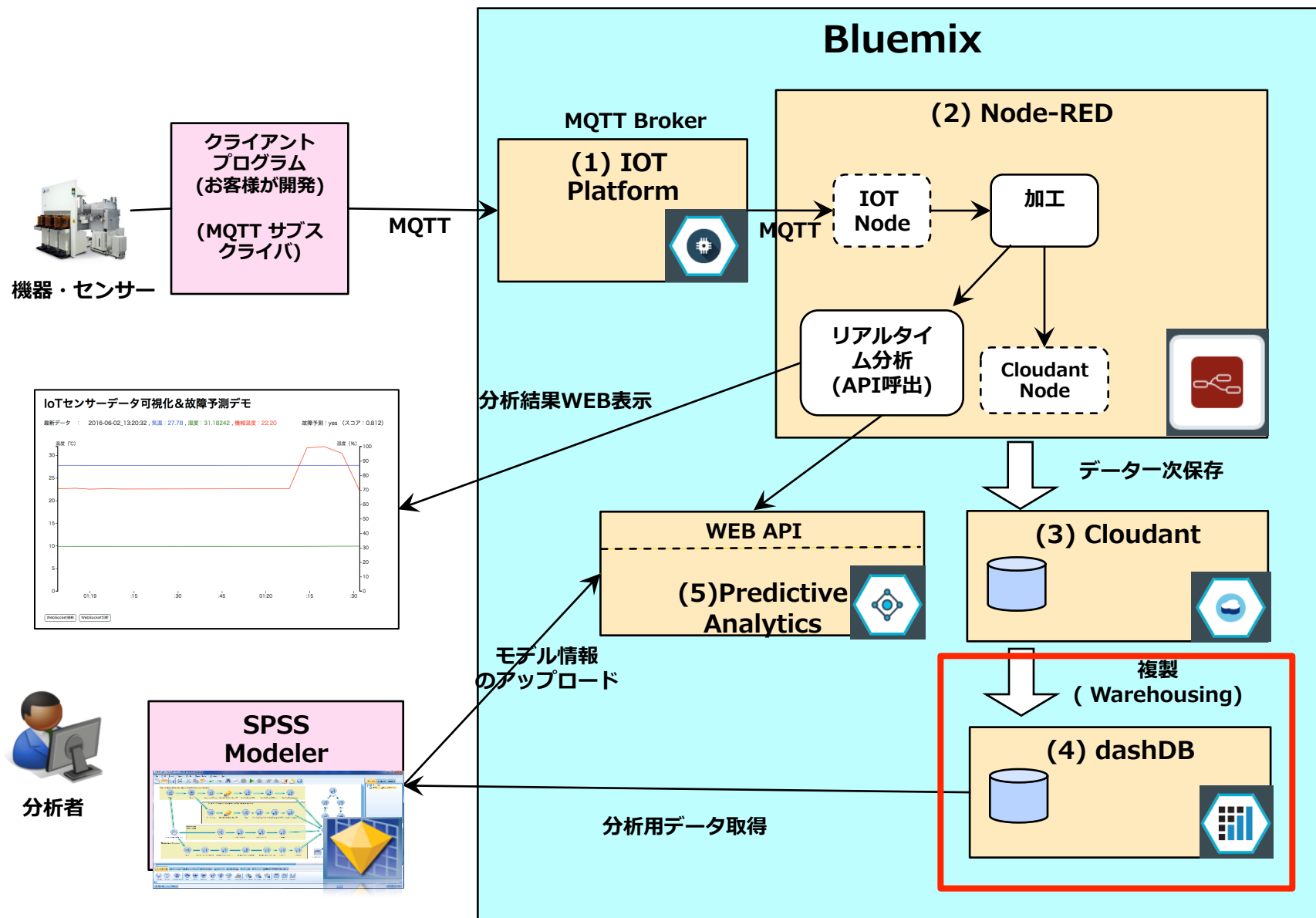
スケールアップや異なるサイト間のデータ同期も容易に実現可能です。

要件により「マルチテナント」「仮想シングルテナント」「物理シングルテナント」を選択可能です。

- ・ **dashDBとの連携**

簡単な設定でほぼリアルタイムにdashDBに対してレプリケーションする事が可能です。(Warehousing)

## (4) dashDB





### dashDBの役割

Cloudant上のIoTデータの複製先として利用される、アナリティクス利用に最適化されたRDBです。

### 特徴

#### ・ DB2 BLUアーキテクチャ

アナリティクス利用に最適なBLUアーキテクチャのRDBです。

カラムオーガナイズ表(データが行でなく列単位で管理されている)でINDEXは必要ありません。

インメモリ最適化機能を持っているので、メモリーを自動的に最適活用し高速アクセスが可能です。

マルチコア対応、SIMD命令セットによる並列処理機能で、処理速度の向上を図っています。

DB2の追加機能であり、アプリケーションからは従来のDB2と同じに見えます。

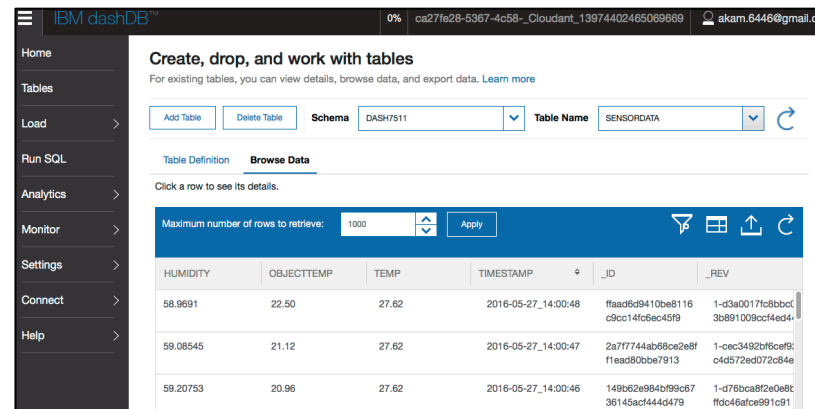
#### ・ 自動運用 (マネージドサービス)

煩雑な運用で必要な作業はIBM側で実施するので、

ユーザー側は特に意識する必要がありません。

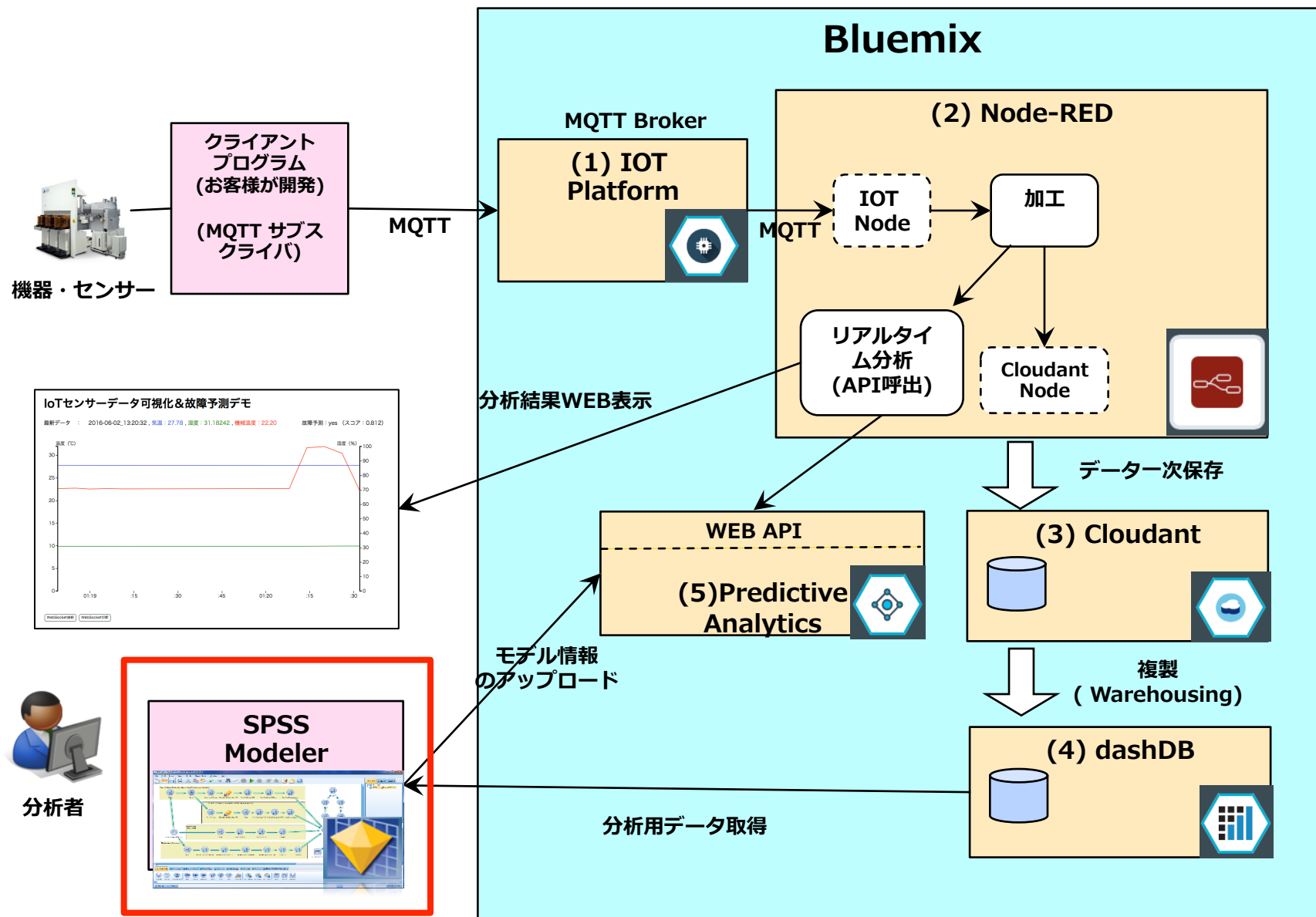
(バックアップ取得・統計情報取得・パラメータ調整等)

プログラム修正適用についてもIBM側で自動的行います。



The screenshot shows the IBM dashDB web interface. The top navigation bar includes 'Home', 'Tables', 'Load', 'Run SQL', 'Analytics', 'Monitor', 'Settings', 'Connect', and 'Help'. The main content area is titled 'Create, drop, and work with tables' and shows a table named 'SENSORDATA' in the 'DASH7511' schema. The table has columns: HUMIDITY, OBJECTTEMP, TEMP, TIMESTAMP, \_ID, and \_REV. The first three columns are highlighted in blue. The table contains three rows of data.

| HUMIDITY | OBJECTTEMP | TEMP  | TIMESTAMP           | _ID                              | _REV                             |
|----------|------------|-------|---------------------|----------------------------------|----------------------------------|
| 58.9691  | 22.50      | 27.62 | 2016-05-27_14:00:48 | f5ad6d9410be8116c9cc14fc6ec45f9  | 1-d3a0017fc0b0c3b891009ccf4ed4   |
| 59.08545 | 21.12      | 27.62 | 2016-05-27_14:00:47 | 2a777744ab88ce2e8ff1ead80bbe7913 | 1-cec3492bfcce9;c4cd572ed072c84e |
| 59.20753 | 20.96      | 27.62 | 2016-05-27_14:00:46 | 149b62e984bf99c6736145ac44449    | 1-d76bca8f2e0e8c;ffc46afce991c91 |





### SPSS Modelerの役割

クラウド上に蓄積されたIoTデータを解析し、予測モデルを作成するツールです。

### 特徴

(SPSS Modeler自体の特徴は他資料を参照して下さい。

ここではCloud連携時の特徴のみ記載します。)

- ・ **Bluemix上のdashDBをデータソースとして利用可能です。**

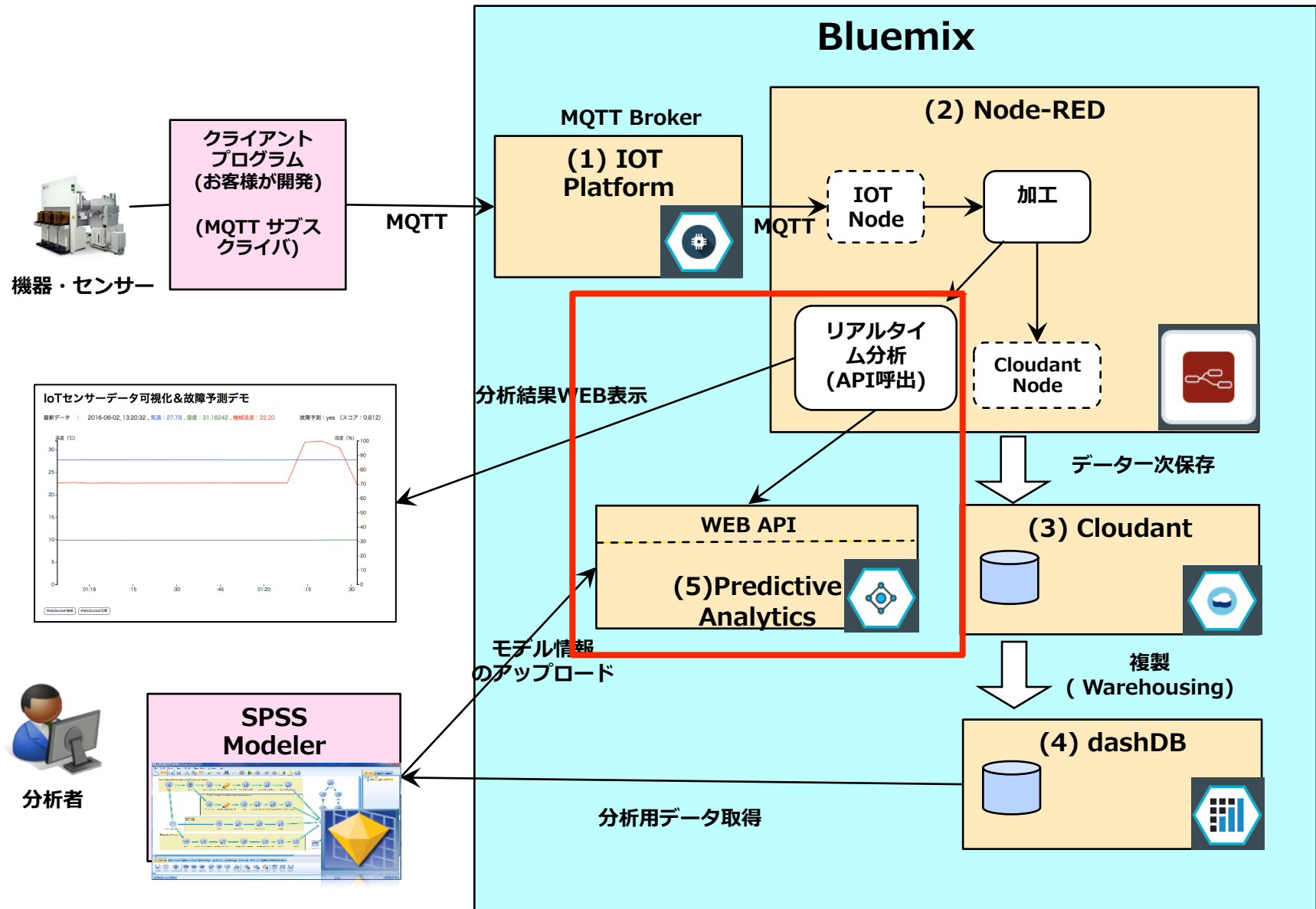
SPSS ModelerからはdashDBはODBC Dataourceとして認識します。SSL通信も可能です。

- ・ **Predictive Analytics連携が可能です。**

Predictive AnalyticsはBluemix上のSPSS Modelerランタイムサービスにあたります。

この連携機能を活用することで、SPSS Modelerで作成した予測モデルをStream Fileとしてexportし、Bluemix上でリアルタイムに実行することが可能です。

# (5) Predictive Analytics





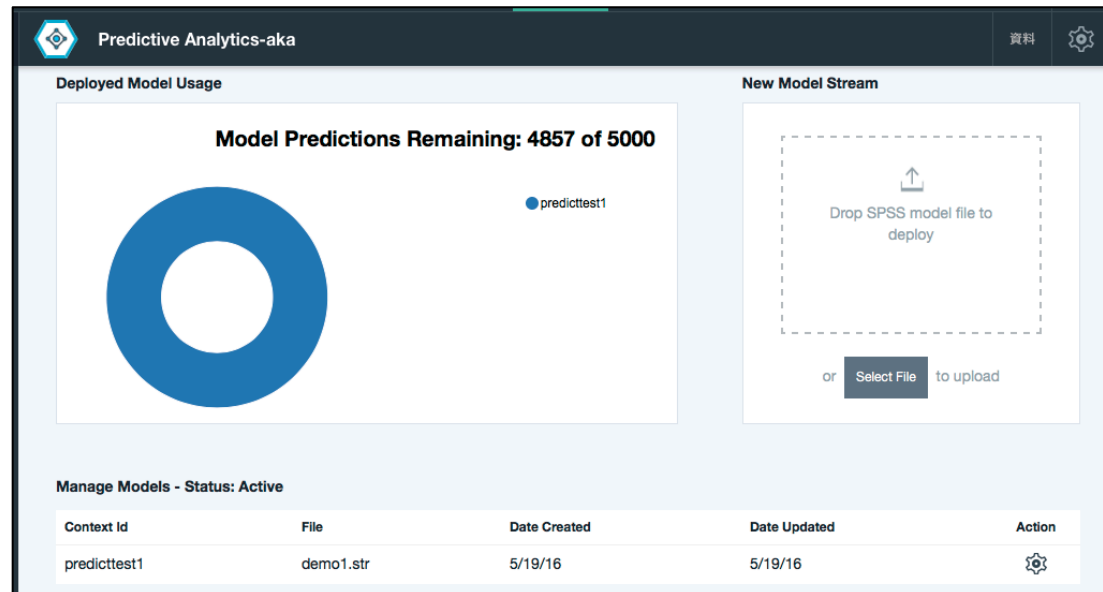


## Predictive Analyticsの役割

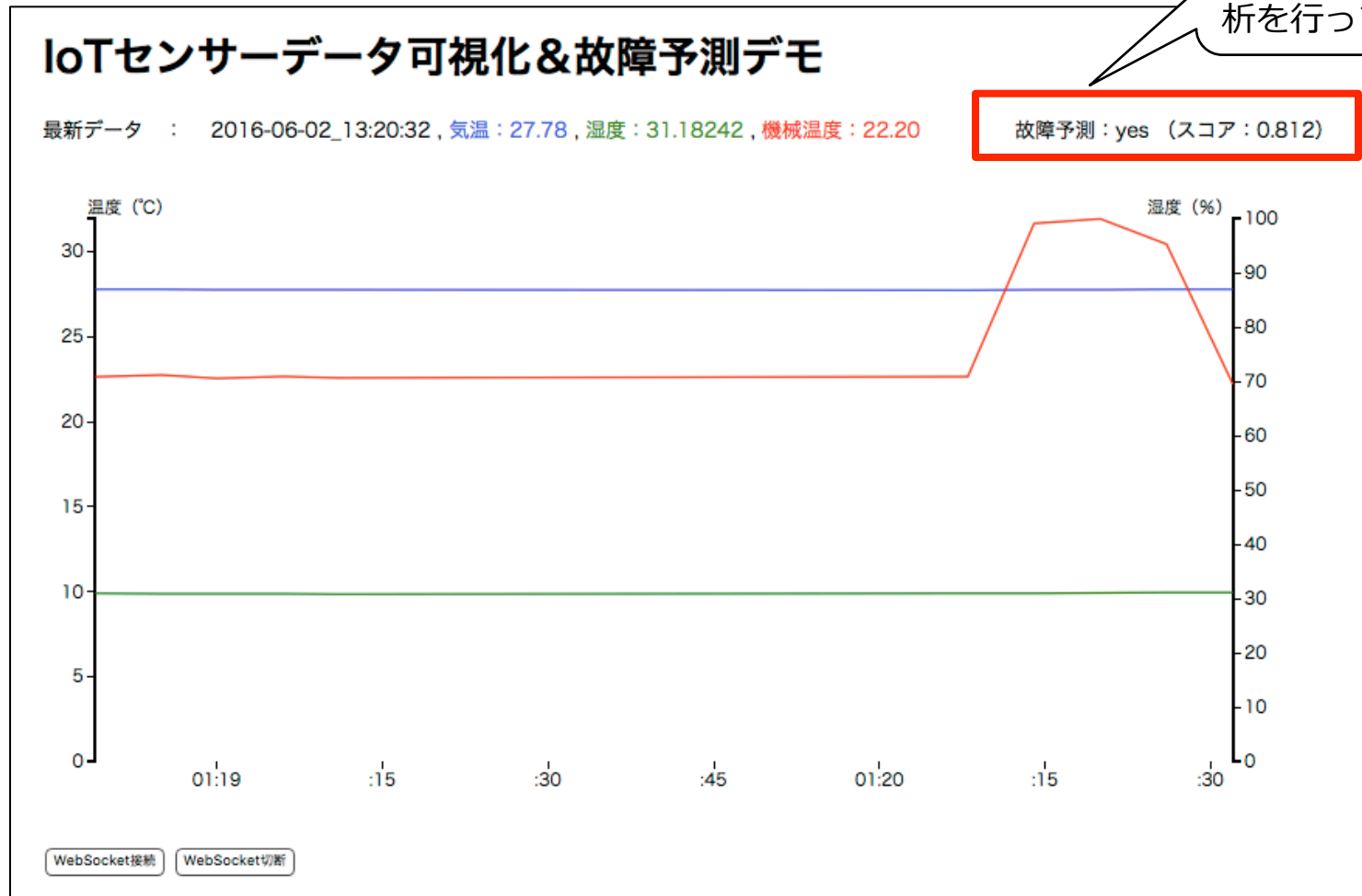
Bluemix上での、予測モデルのリアルタイム実行基盤です。

## 特徴

- ・ Bluemix上のサービスとして稼働します。
- ・ モデルとしてはSPSS Modelerで作成した予測モデルをuploadして利用します。
- ・ REST API形式で呼び出しを行います。

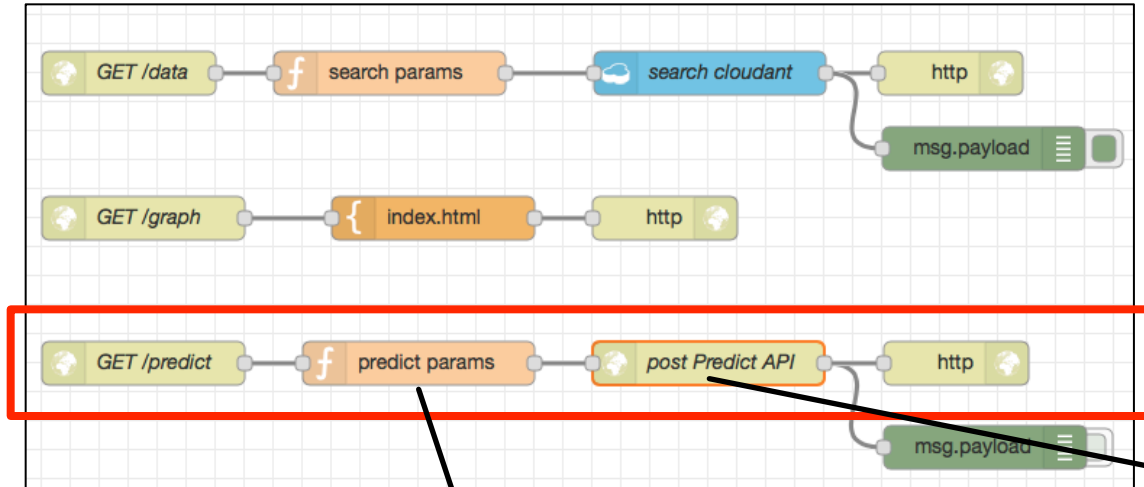


## リアルタイム分析を行うサンプルアプリの例



リアルタイムにSPSSで作った予測モデルを使って分析を行っている

Node-REDからPredictive Analytics機能の呼び出しを行っている部分



Function

```
1 msg.payload = {
2   "tablename": "SECOM_DEMO_DATA.csv",
3   "header": ["seq_no", "failure", "temperature", "humidity", "objecttemperature"],
4   "data": [
5     ["test1", "",
6      msg.payload.param_temp,
7      msg.payload.param_hum,
8      msg.payload.param_objt
9     ]
10  ]
11 }
12 return msg;
```

パラメータ組立て

|                           |  |
|---------------------------|--|
| Method                    | POST   |
| URL                       | https://palbyp.pmservice.ibmcloud.com/pm/v1/score/predicttest1?accesskey=U |
| Use basic authentication? | <input type="checkbox"/>   |
| Return                    | a parsed JSON object   |
| Name                      | post Predict API   |

API呼び出し(Rest)