

DX時代における、AWSデータ活用手法

～データ連携ツールでAWSをノーコード開発で 高速につなぎ、フル活用する！

アステリア 株式会社

エンタープライズ本部 マーケティング部 部長

東出 武也

2019.05.23

- ✓ 東証 1 部上場 (3853)
- ✓ 1998年9月創立
- ✓ 「つなぐ」ソフトウェア製品の開発・販売
- ✓ 世界 5ヶ国に拠点

英国

中国

日本

米国

シンガポール

- ✓ 主力製品の「ASTERIA Warp」「Handbook」は日本国内市場でシェアNo.1

- ✓ 2018年10月に「インフォテリア株式会社」から「アステリア株式会社」に商号変更

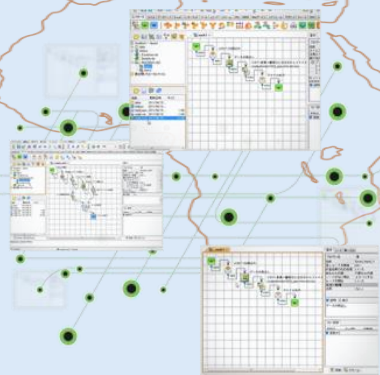


代表取締役
執行役員社長/CEO
平野 洋一郎

アステリア 製品/サービス

戦略的システム連携で
ビジネスの明日を想像する

**asteria
warp**



2002年～

シェアNo.1のMCM、
さらに飛躍。



Handbook



2009年～

ビジネス・アプリを手軽に作ろう！

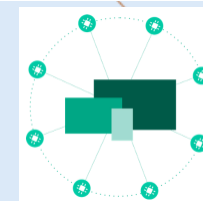
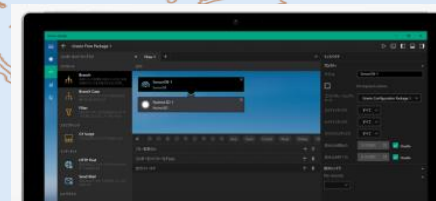
Platio



2017年2月～

エッジコンピューティング用ミドルウェア

Gravio

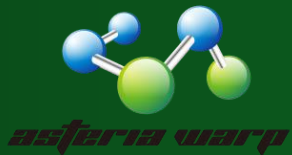


2017年6月～



Agenda

- ❖ DX時代におけるデータ連携の必要性と連携パターン
- ❖ データ連携で押さえるべきポイント
- ❖ ASTERIA Warpを利用したデータ活用手法
- ❖ AWS連携機能
- ❖ AWS連携パターンとDemo



DX時代における データ連携の必要性和連携パターン

データ連携の必要性和連携パターン

✓ DX(デジタルトランスフォーメーション)

「ITの浸透が、人々の生活をあらゆる面でより良い方向に変化させる」という概念: ストルターマン教授(出展: Wikipedia)

DXの段階

- 第1フェーズ: IT利用による業務プロセスの強化
- 第2フェーズ: ITによる業務の置き換え
- 第3フェーズ: 業務がITへ、ITが業務へとシームレスに変換される状態

データ連携の必要性と連携パターン

DXの段階

第1フェーズ

IT利用による業務プロセスの強化

標準化された業務プロセスを
情報システムに置き換える

業務システム

ERP

SFA/CRM



第2フェーズ

ITによる業務の置き換え

ITで業務を代替させ、自動化させる

RPA

データ連携



BizRobo!

WinActor®

× **asteria warp**
CORE

第3フェーズ

業務がITへ、ITが業務へとシームレ
スに変換される状態

ITと業務の現場が一体となって、改
善活動を高速で繰り返しながら、常
に最適な状態を維持し、業務を遂行
する

IoT

AI

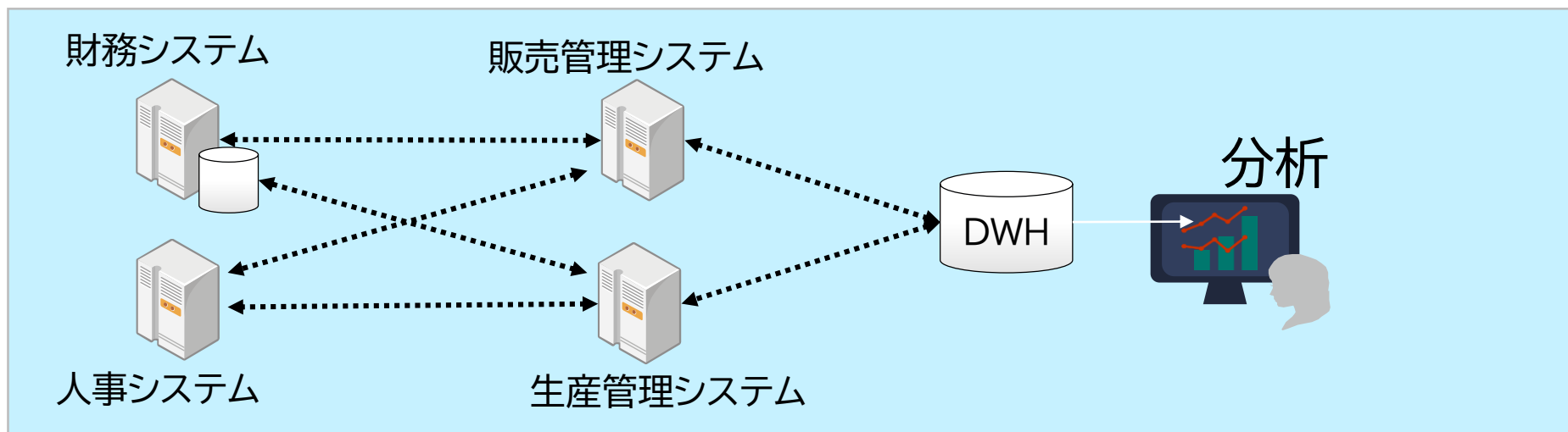
AWS

Google/Microsoft

データ連携の必要性と連携パターン

✓ 以前までは

- 分散されたシステム間のデータ連携
- BI・ダッシュボードによる分析
etc.



データ連携の必要性と連携パターン

✓ 現状は

- クラウド移行によるハイブリッドクラウド連携
- クラウド to クラウド
- IoTなどのビッグデータ活用(社内データ資産とのマッシュアップ)
- AI活用 etc.



データ連携の必要性と連携パターン

✓ よって

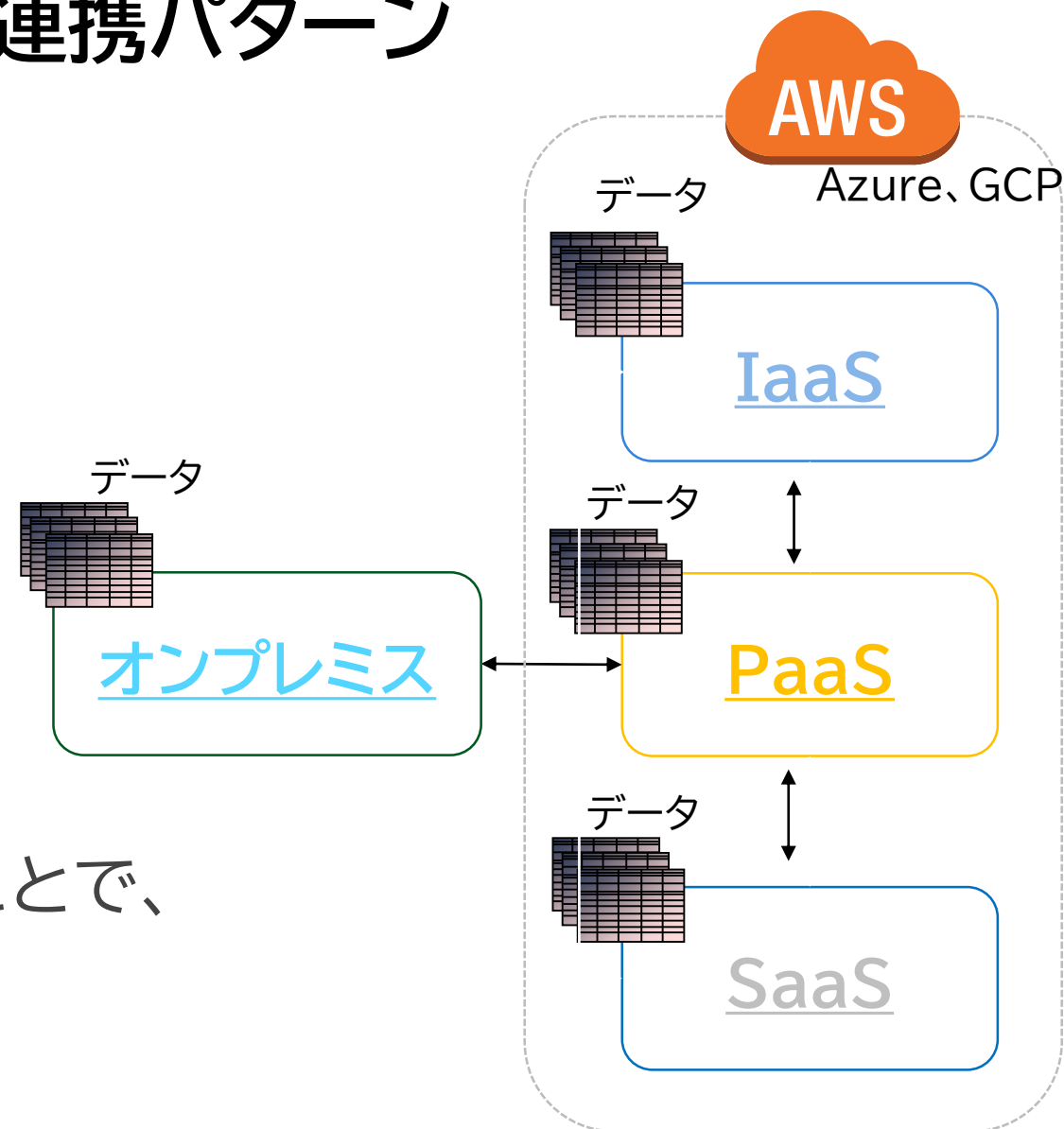
“データのみがIT資産”になる

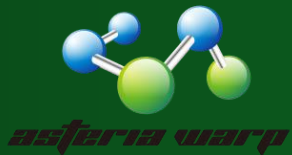


今後はデータを柔軟に連携させることで、

“データの価値”を高め、

DXをさらに推進する必要がある





データ連携で押さえるべきポイント



データ連携で押さえるべきポイント

API を攻略する！

API : Application Program Interface



ほとんどのクラウドサービスやパッケージ製品が
Web API を公開

データ連携で押さえるべきポイント

- インターフェース(Web API)

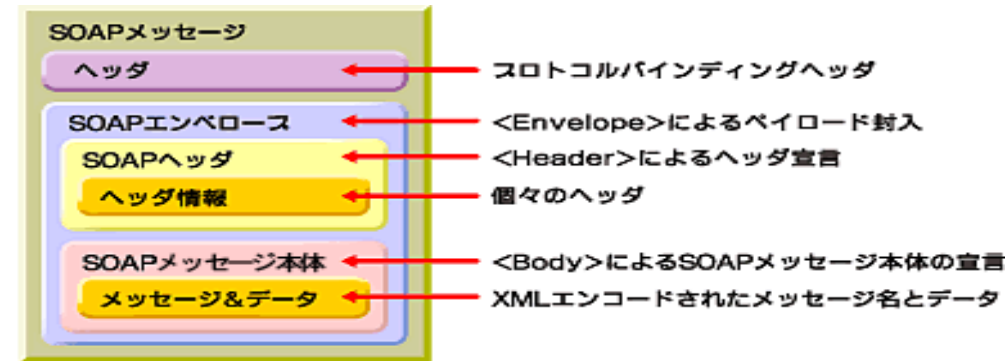
ほとんどのサービスが

SOAP (Simple Object Access Protocol)

or

REST (Representational State Transfer)

を提供している



SOAP vs. REST

- ✓ 現在ではよりシンプルなREST APIが事実上の標準
- ✓ SOAPはRESTに比べてデータ構造が複雑
- ✓ 入力パラメータが単純であればRESTで対応しやすい

データ連携で押さえるべきポイント

- データフォーマット(Web API)

ほとんどのサービスが

XML (Extensible Markup Language)

or

JSON (JavaScript Object Notation)

で提供

リスト1:XMLの例

```
<employees>
  <employee>
    <empId>000001</empId>
    <department>企画部</department>
    <name>山田 太郎</name>
  </employee>
  <employee>
    <empId>000002</empId>
    <department>営業部</department>
    <name>山田 次郎</name>
  </employee>
</employees>
```

リスト2:JSONの例

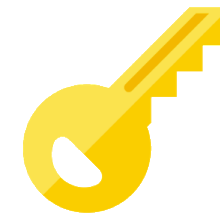
```
[
  {
    "empId": "000001",
    "department": "企画部",
    "name": "山田 太郎",
  },
  {
    "empId": "000002",
    "department": "営業部",
    "name": "山田 次郎",
  }
]
```

XML vs. JSON

- ✓ XMLでは表記の冗長さに加え、DTDやXML スキーマ、名前空間などさまざまな事柄を意識する必要がある
- ✓ JSONは非常にシンプルで覚えることが少ない

データ連携で押さえるべきポイント

- 認証(Web API)
 - ・ AWS
 - AuthorizationヘッダとHMACハッシュ署名による認証
 - ・ Google / Microsoft Dynamics / Salesforce / Facebook / Twitter
 - OAuth1/OAuth2
 - ・ kintone
 - Authorizationヘッダ + Basic認証
 - etc.



Web APIによる認証方式は種類が多い...



データ連携で押さえるべきポイント

- 既存システムとの連携

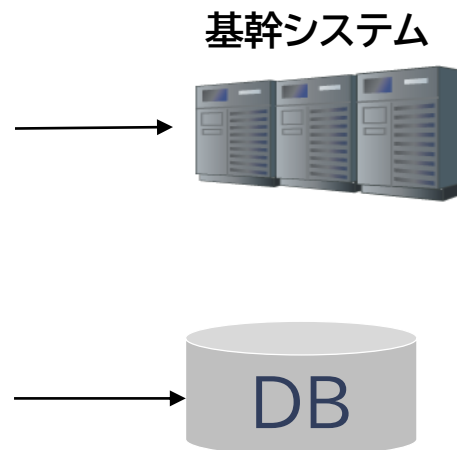
相手側のAPIにあわせ、従来通りの手段で連携

- ファイル連携

- FTP/MQ

- DB連携

- JDBC/ODBCなど



基本、相手側システムやパッケージのインターフェースに従う

BAPI/RFC/IDOC

SAP/R3

データ連携で押さえるべきポイント

✧ クラウド連携で注意すべき点

- ✓ データ転送(参照)に時間が掛かる
ログファイル転送とかは1ファイルが数GByte以上にも
 - 圧縮、分割転送、専用線
- ✓ トランザクションがきかない
SOAP/RESTなどはPUTしたらコミット
 - 設計でカバー、クラウドDB利用
- ✓ 利用制約がある
クエリの発行可能総数、一度の抽出件数/登録件数など
 - 設計でカバー、プラン変更



データ連携で押さえるべきポイント

データ連携部分の開発手法は？

- プログラミング言語を使った場合のロジック
 - インターフェース数が少なければプログラミングで...
 - 専用ライブラリも用意されているので頑張れば...
 - できるだけコストをかけたくないから...

でも...

- ✓ スケジューラ機能は？
- ✓ 大量データ処理は？リカバリは？
- ✓ API変更...
- ✓ メンテナンスは？連携先変わったら作り直し？

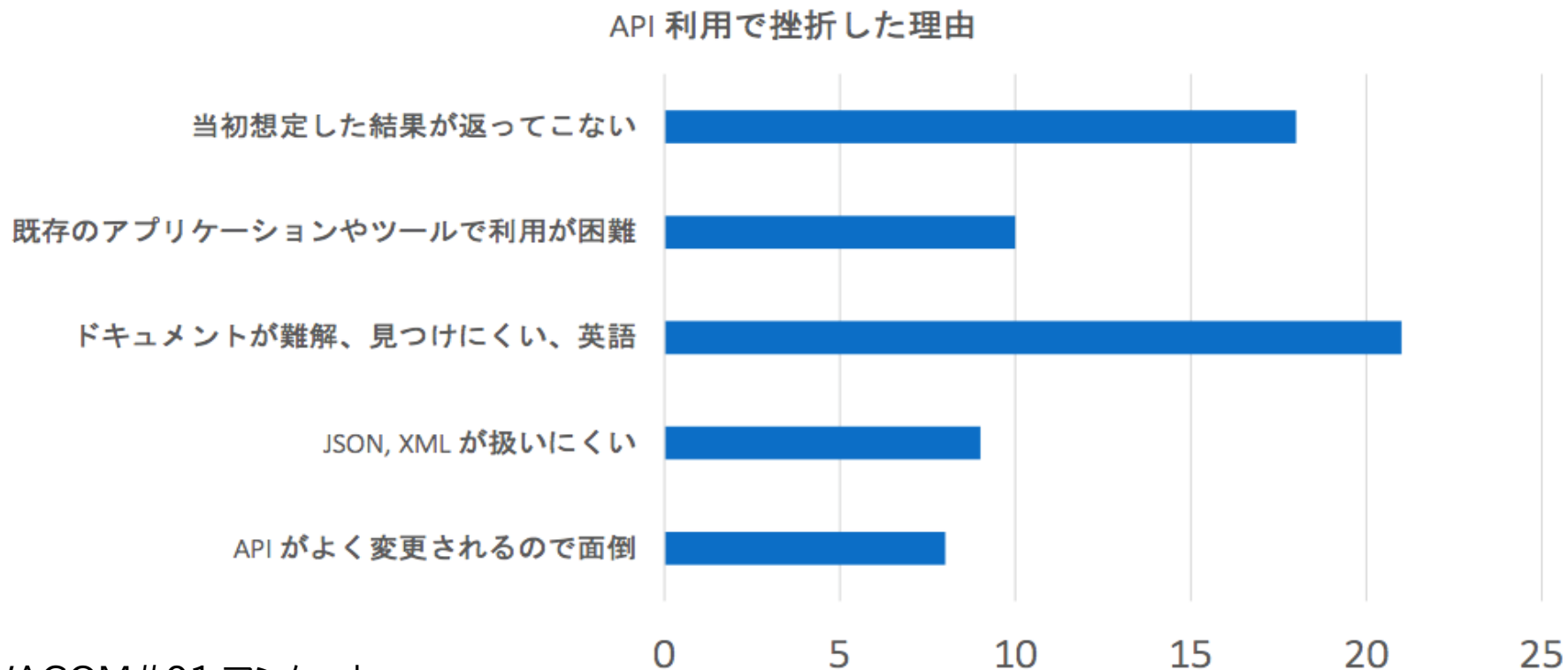
そして、現実的な課題としては...



データ連携で押さえるべきポイント

Japan Web API Community での調査結果

□ API 連携開発で挫折した理由は？



出展:JWACOM#01 アンケート

データ連携で押さえるべきポイント

- APIを使ってみる際の課題

リファレンス
が難解



認証も難解
(トークンの有効期間・・・)



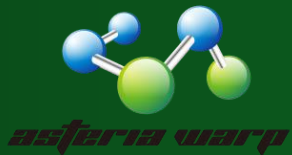
プログラム
工数



すべてのAPIで異なり、知識があまり生かせない



データ連携ツール「ASTERIA Warp」で解決します！



ASTERIA Warpを利用したデータ活用手法

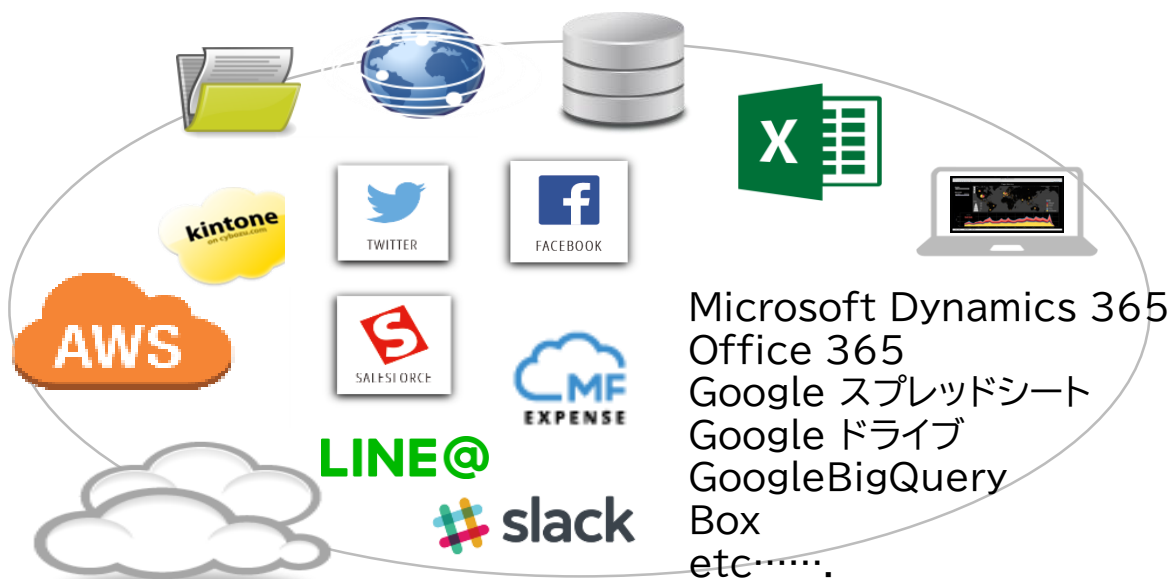




ASTERIA Warp とは

ASTERIA Warpは、“ノン・コーディング”で設計開発を行うことで、様々なシステムやサービスと連携し、業務の効率化やデータの活用を実現するデータ連携ツールです。

70以上のサービスやシステムと迅速に連携



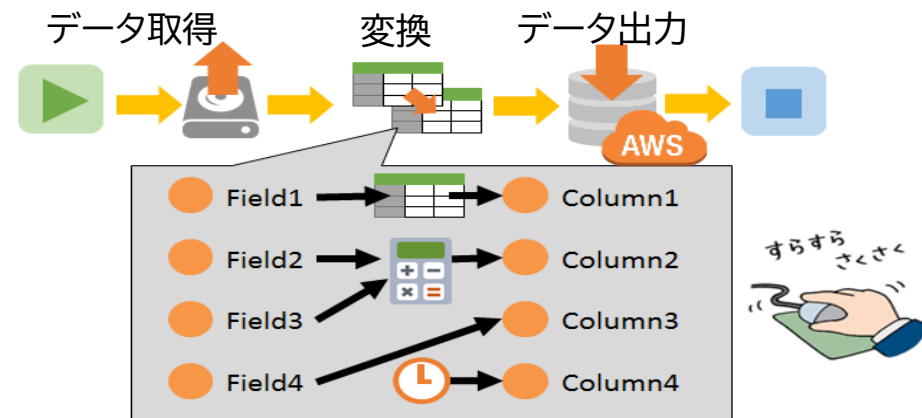
シンプルな操作で超高速開発



スケジュール起動
外部トリガー起動



月額3万円から



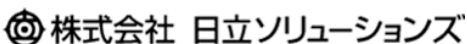
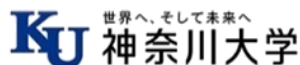
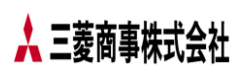
アイコンを並べて
ノン・コーディングで
データ連携を定義します



導入実績 7,783社(2019 年3月末現在) & 12年連続シェアNo.1



テクノ・システム・リサーチ
「2018年 ソフトウェアマーケティング
総覧 EAI/ESB市場編」(数量ベース)





70種類以上の豊富な接続先

データベース

- Oracle
- Microsoft SQL Server
- Microsoft Azure SQL Database
- MySQL
- Amazon RDS
- Amazon Athena
- IBM Db2
- Microsoft Access
- Microsoft Azure Cosmos DB
- MariaDB
- PostgreSQL
- Amazon DynamoDB
- MongoDB
- Google Cloud SQL
- FileMaker

DWH / ビッグデータ

- Amazon Redshift
- IBM PureData
- Google BigQuery
- Amazon EMR

クラウド (IaaS)

- Microsoft Azure
- Google Cloud Platform
- Cloud[®]
- Amazon Web Services
- さくらのクラウド
- Oracle Cloud Platform

ファイルフォーマット

- CSV
- ZIP
- HTML
- テキスト
- XML
- GZIP
- MIME (S/MIME)
- バイナリ
- PDF
- TAR
- JSON
- 固定長

ネットワーク

- SOAP
- FTP / SFTP
- SMTP / SMTPS
- REST
- POP3 / POP3S
- LDAP
- HTTP / HTTPS
- IMAP / IMAPS

クラウド ストレージ

- Amazon S3
- Google ドライブ
- Google Cloud Storage
- Microsoft OneDrive
- Box

SNS

- Facebook
- Slack
- Twitter
- LINE
- Amazon SNS

オフィス/グループウェア

- Microsoft Excel
- Microsoft SharePoint
- Google コンタクト
- クリプト便
- Microsoft Active Directory
- kintone
- Google カレンダー
- D³Worker
- Microsoft Office 365
- IBM Notes/Domino
- Google スプレッドシート
- POWER EGG

会計

- 請求管理ロボ
- MFクラウド経費

BPM / BRMS / ワークフロー

- BIZ PLATFORM
- EXPLANNER/FL
- intra-mart
- InnoRules

SFA / CRM

- Microsoft Dynamics CRM (365)
- Salesforce
- クライゼル

ERP

- SAP ERP
- Microsoft Dynamics NAV (365)
- Microsoft Dynamics AX (365)
- IFS Applications
- ServAir
- NetSuite
- RobotERP ツバイン
- スーパーカクテルデュオ

EDI

- JFT/SaaS
- Biware

BI

- Tableau
- Dr.Sum EA

マーケティング

- Google Analytics
- Oracle Eloqua
- Sansan
- Google 広告
- uSonar
- 名刺de商売繁盛
- Marketo
- Data-Master

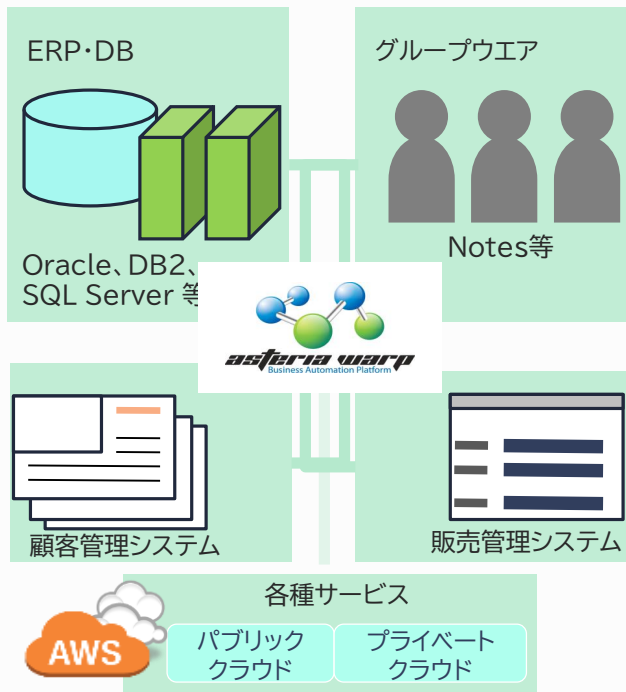
AI/RPA/OCR

- Amazon Rekognition
- Google Cloud Vision
- Azure Cognitive Services
- BizRobo!



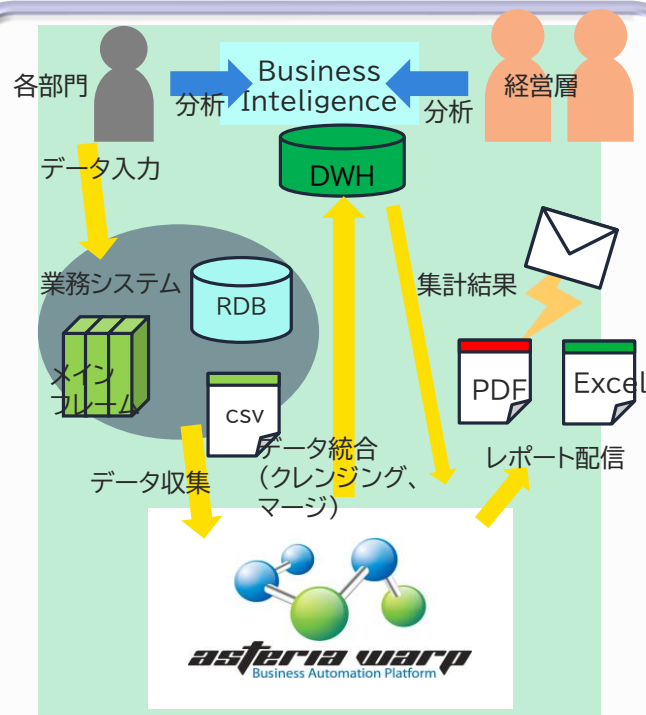
利用シーン

①システム連携基盤 (EAI・ESB・ハイブリッドクラウド)



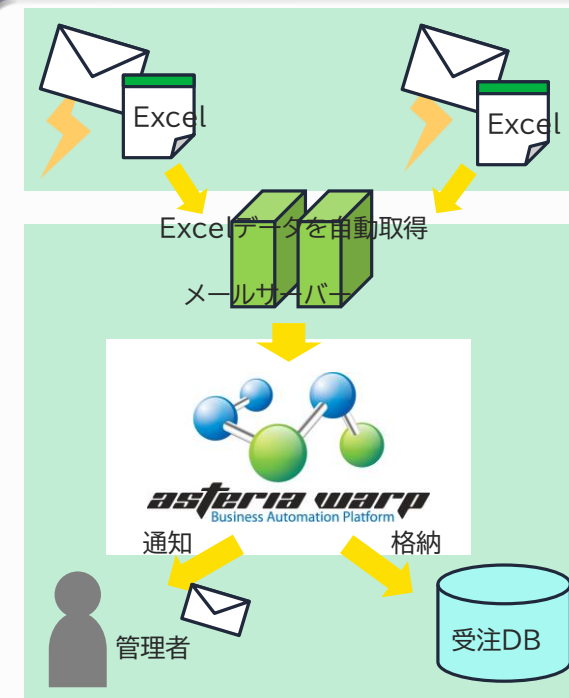
- 社内外に散在する各システムの疎結合連携を実現
- 多重入力を無くし業務効率向上が可能

②データ統合基盤 (ETL)



- 煩雑なデータ収集・分析業務の自動化を実現
- 高速処理機能によりDWHへの高速Loadを実現

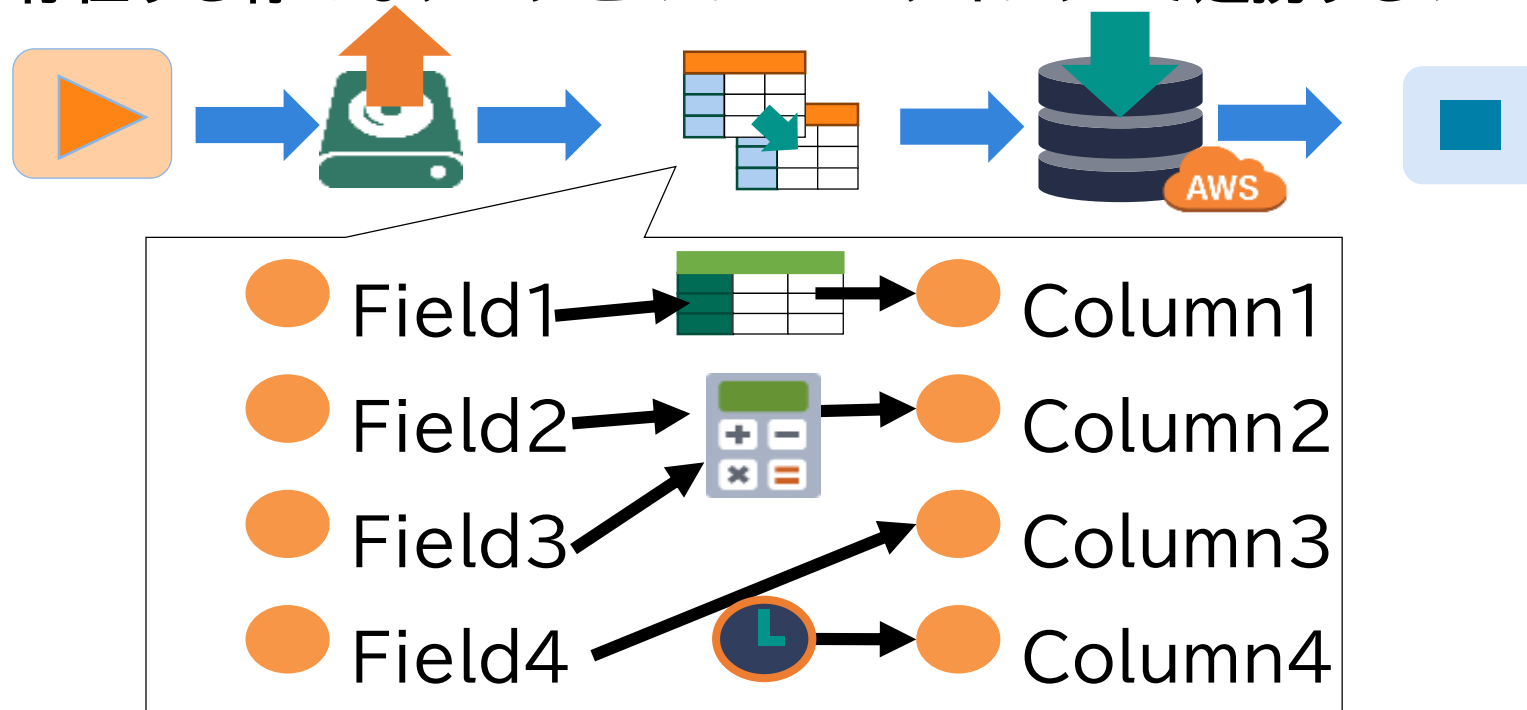
③メールやExcelとの連携 (簡易EDI)



- 手作業で行っている受発注処理の自動化を実装可能
- メール送受信からExcelデータからの抽出や書き込みが可能

ノンコーディングで高速開発

ASTERIA Warp は社内には存在する各種システムやデータ、クラウドサービス上に存在する様々なデータを“ノン・コーディング”で連携するツールです。



ASTERIA Warp は各種の機能を内包したアイコンを並べてデータ連携を定義します。

ASTERIA Warp とプログラミング(手組)

プログラミング(手組)

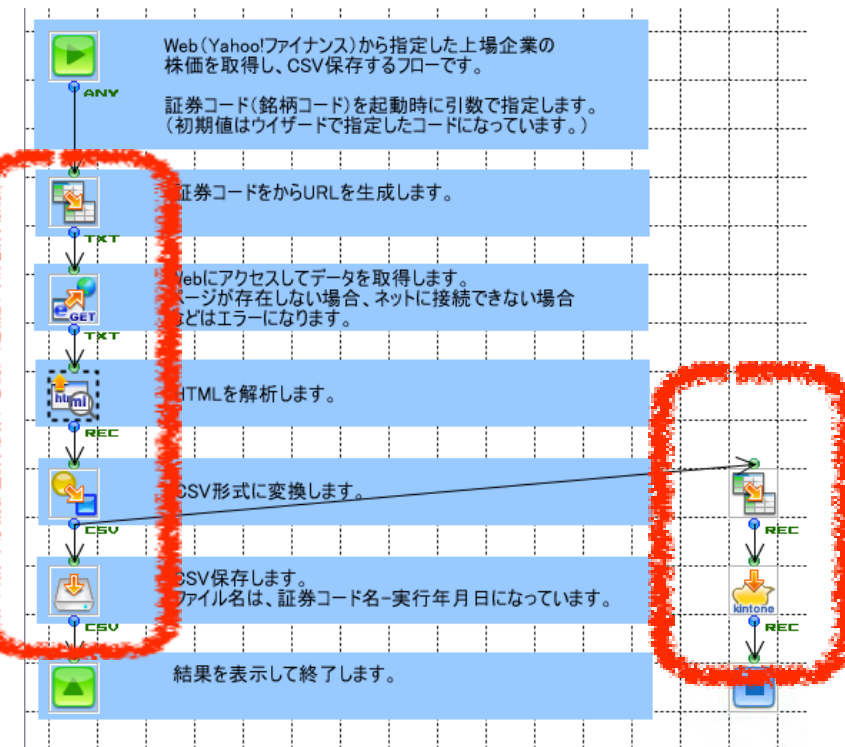
```

1 var client = require('cheerio-httpcli');
2
3 // 株価取得
4 client.fetch('http://stocks.finance.yahoo.co.jp/stocks/history?code=3853', '', function (err, $, res) {
5   var jsonStr = '{"app": 37, "records": [';
6   $('table.boardFin tr:nth-of-type(n+2)').each(function (idx) {
7     $(this).each(function (idx2) {
8       var dateStr = $('td:nth-of-type(1)', this).text().split(/[年|月|日|]/).join('-');
9       dateStr = dateStr.substr(0, dateStr.length-1);
10      jsonStr = jsonStr + '{"日付": "' + dateStr + '",';
11      jsonStr = jsonStr + '"始値": ' + parseInt($('td:nth-of-type(2)', this).text().split(',').join('').trim()) + ',';
12      jsonStr = jsonStr + '"高値": ' + parseInt($('td:nth-of-type(3)', this).text().split(',').join('').trim()) + ',';
13      jsonStr = jsonStr + '"安値": ' + parseInt($('td:nth-of-type(4)', this).text().split(',').join('').trim()) + ',';
14      jsonStr = jsonStr + '"終値": ' + parseInt($('td:nth-of-type(5)', this).text().split(',').join('').trim()) + ',';
15      jsonStr = jsonStr + '"出来高": ' + parseInt($('td:nth-of-type(6)', this).text().split(',').join('').trim()) + ',';
16      jsonStr = jsonStr + '"調整後終値": ' + parseInt($('td:nth-of-type(7)', this).text().split(',').join('').trim()) + '},';
17    });
18  });
19  jsonStr = jsonStr.substr(0, jsonStr.length-1);
20  jsonStr = jsonStr + ']}';
21  optionStr = '{"uri": "https://infoteria-demo.cybozu.com/k/v1/records.json", ' +
22    '"headers": {"Content-type": "application/json", ' +
23    '"X-Cybozu-Authorization": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"}, ' +
24    '"json": ' + jsonStr + '}'';
25  var option = JSON.parse(optionStr);
26
27  // kintone 書き込み
28  var request = require('request');
29  request.post(option, function(error, response, body){
30    if (!error && response.statusCode == 200) {
31      console.log("success");
32    } else {
33      console.log(error);
34    }
35  });
36 });

```

ノン・コーディング

VS



数十分～数時間

数分

プログラミング(手組)

```

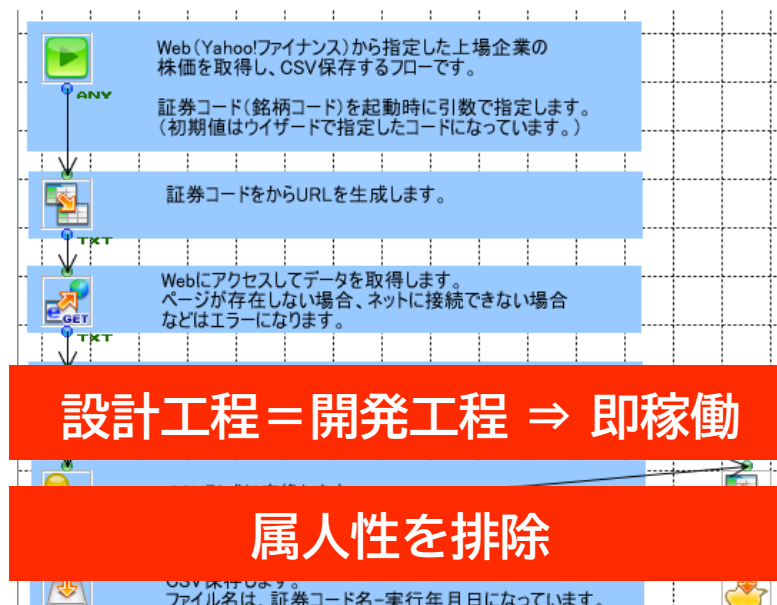
1 var client = require('cheerio-httpcli');
2
3 // 株価取得
4 client.fetch('http://stocks.finance.yahoo.co.jp/stocks/history?code=3853','', function (err, $, res) {
5   var jsonStr = '{"app": 37,"records": [';
6   $('table.boardFin tr:nth-of-type(n+2)').each(function (idx) {
7     $(this).each(function (idx2) {
8       var dateStr = $('td:nth-of-type(1)',this).text().split(/[年|月|日|/]).join('-');
9       dateStr = dateStr.substr(0,dateStr.length-1);
10      jsonStr = jsonStr + '{"日付": "' + dateStr + '",';
11      jsonStr = jsonStr + '"始値": ' + parseInt($('td:nth-of-type(2)',this).text().split(',').join('').trim()) + ',';
12      jsonStr = jsonStr + '"高値": ' + parseInt($('td:nth-of-type(3)',this).text().split(',').join('').trim()) + ',';
13      jsonStr = jsonStr + '"安値": ' + parseInt($('td:nth-of-type(4)',this).text().split(',').join('').trim()) + ',';
14      jsonStr = jsonStr + '"終値": ' + parseInt($('td:nth-of-type(5)',this).text().split(',').join('').trim()) + ',';
15      jsonStr = jsonStr + '"出来高": ' + parseInt($('td:nth-of-type(6)',this).text().split(',').join('').trim()) + ',';
16      jsonStr = jsonStr + '"調整後終値": ' + parseInt($('td:nth-of-type(7)',this).text().split(',').join('').trim()) + '},'
17    });
18  });
19  jsonStr = jsonStr.substr(0,jsonStr.length-1);
20  jsonStr = jsonStr + ']}';
21  optionStr = '{"uri": "https://infoteria-demo.cybozu.com/k/v1/records.json", ' +
22    '"headers": { "Content-type": "application/json", ' +
23    '"X-Cybozu-Authorization": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx", ' +
24    '"json": ' + jsonStr + '}}';
25  var option = JSON.parse(optionStr);
26
27 // kintone 書き込み
28 var request = require('request');
29 request.post(option, function(error, response, body){
30   if (!error && response.statusCode == 200) {
31     console.log("success");
32   }
33 });

```

→65~100stepのjava
 サンプルの検索、コピー＆ペースト、
 コンパイル、テスト、仕様書作成など、
 タスクは山積み。

ノン・コーディング

VS.



→アイコンを並べるだけのカンタン開発
 詳細設計工程と開発工程を同時化により
 開発工数を削減
 仕様書も自動作成

さらにテンプレート機能で開発不要に

フローテンプレートを利用することで、開発がほぼ不要に！
ナビゲーション画面で設定することで自動で処理フローが生成



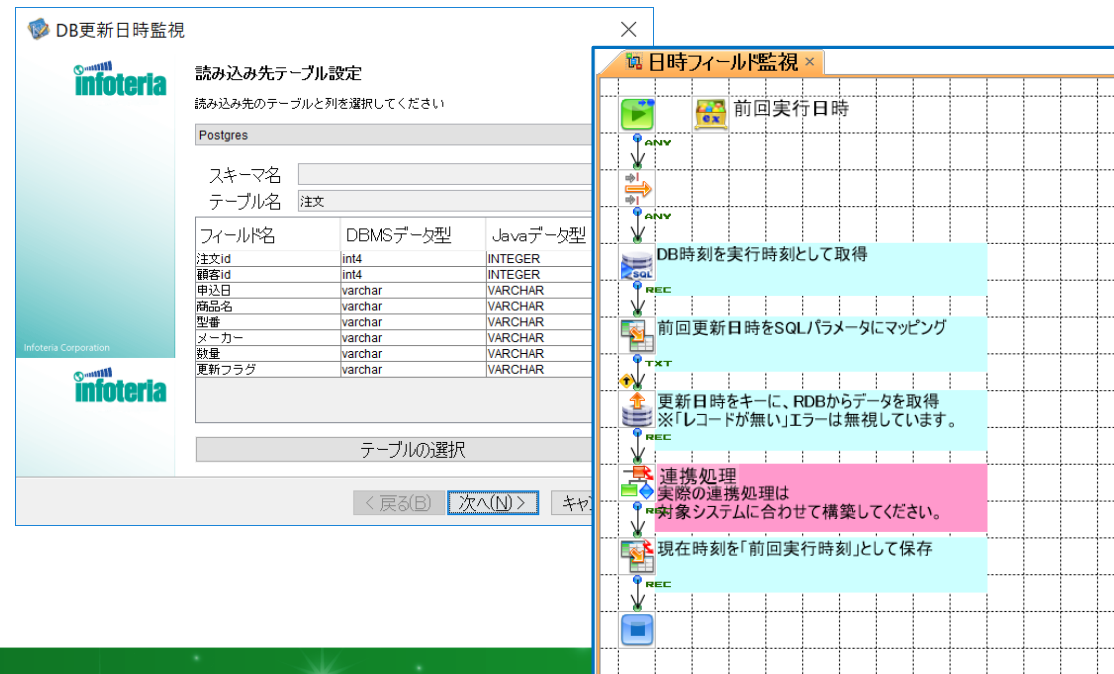
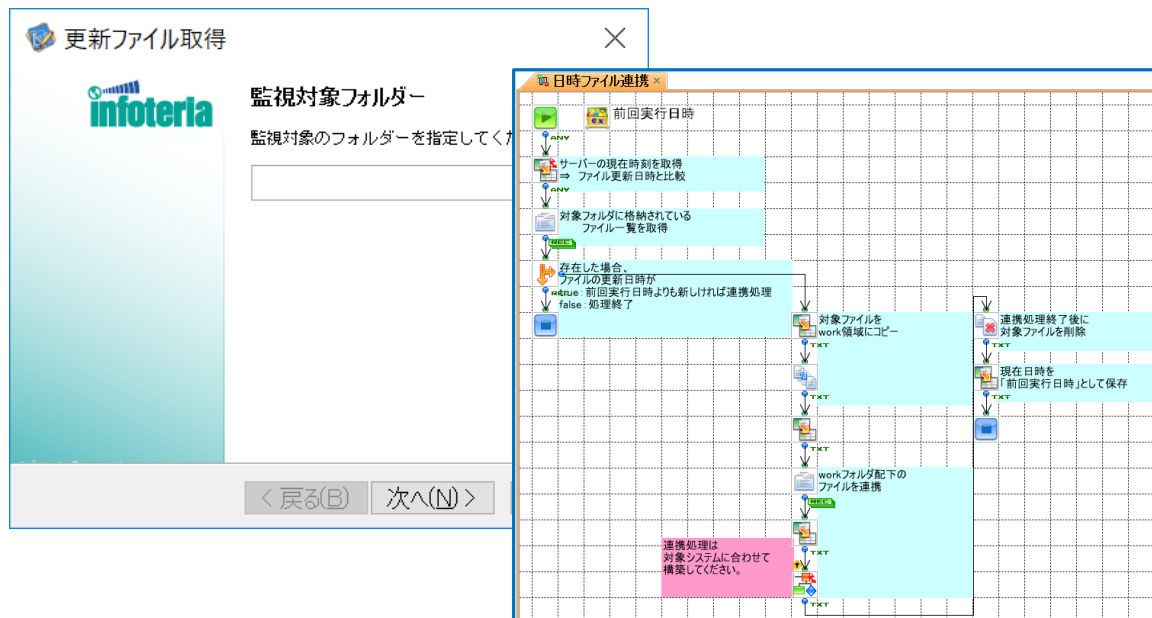
更新ファイル取得

フォルダーでファイルの更新日時を監視し更新されたファイルを取得



DB更新日時監視

データベースの更新日時フィールドを監視し、前回実行時以降に更新されたレコードを抽出



充実したエンタープライズ機能

チェックポイント機能

- 中断してしまったフローを途中から再開(リトライ)
- 障害で中断した長時間処理の再開に効果的
- 別サーバでのリカバリも可能



優先実行モード

- フローの実行に「優先」モードを追加
- 通常のフローが実行中でも優先して実行される
- Web系の処理とバッチ系の処理が混在している環境など、優先度が決められる場合に有効



大量データ処理

- APIを利用したバルクでの登録機能
- 並列処理機能でさらにパフォーマンス向上
- 処理フロー単位でのトランザクションが可能

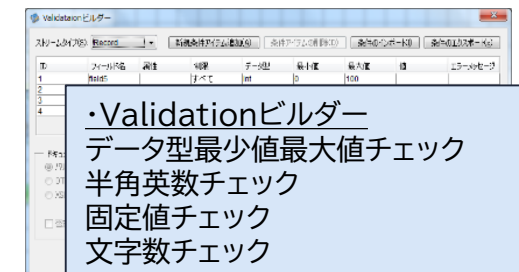
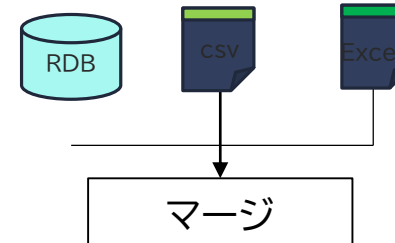


プロパティ名	値
名前	RDBPut1
コネクション名	OracleXe
実行する処理	Insert
テーブル名	HR.EMPLOYEES
処理件数	
バッチ処理件数	100000
タイムアウト(秒)	0
エラー処理	
汎用	(なし)
接続エラー	(なし)



マージ、チェック処理

- 仮想テーブル機能を利用したマージ処理
- 関数機能を利用したコード変換処理
- Validation機能でのデータチェック



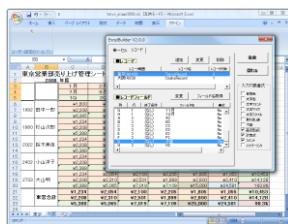
開発生産性を向上する支援機能

開発支援ツール

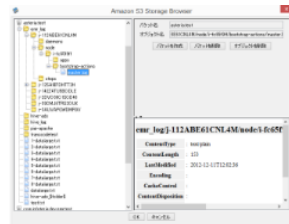
- データベースから読み込むテーブル、フィールドをGUI形式のツール「SQLビルダー」で選択
- 実際に読み込むExcelブック、書き込むExcelブックをもとに「Excelビルダー」のGUIで設定



SQLビルダー



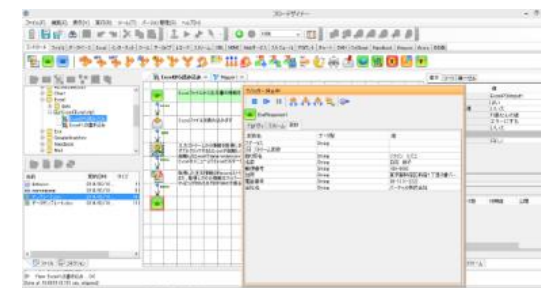
Excelビルダー



AWS S3ブラウザ

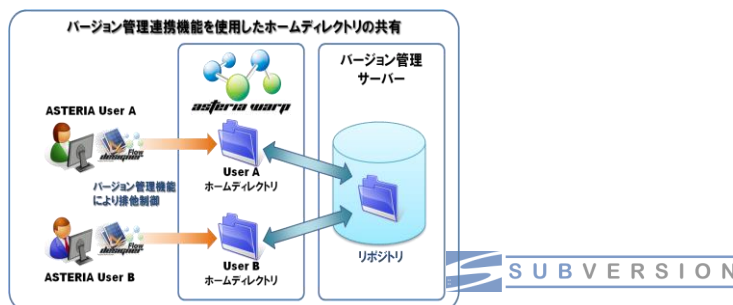
デバッグ実行

- ブレークポイントをフローに設定することでリアルタイムにデバッグが可能
- 処理を順に追って変更過程の処理を検証し修正可能



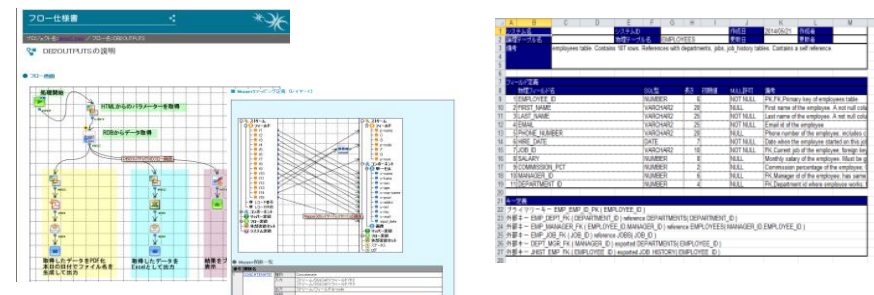
バージョン管理

- Subversionとの連携
- プロジェクト共有、変更履歴管理などチーム開発支援



仕様書出力(フロー、テーブル)

- 詳細なフロー定義情報、DBテーブル情報を出力
- 差分情報も出力可能



まとめ

- APIを使ってみる際の課題

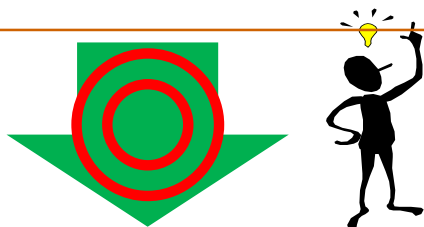
リファレンス
が難解



認証も難解
(トークンの有効期間..)



プログラム
工数



API仕様は専用アダプター
が吸収



認証も専用アダプターが
吸収



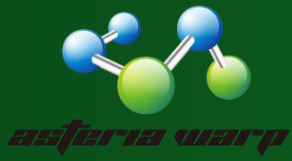
ノン・コーディング
での超高速開発！

まとめ

- データ連携比較（データ連携ツール vs. プログラミング手組）

	データ連携ツール (ASTERIA Warp)	プログラミング (手組)
開発生産性(スピード)	◎ ノン・コーディング	○
汎用性	○	◎
メンテナンス(拡張)性	◎ 処理が可視化	×
アダプター機能(認証含む)	◎ アダプター豊富、API変更もサポート	△
スケジューラ	◎	△(別製品が必要)
運用管理機能	○	△(別製品が必要)
保守(サポート)	◎	△
コスト	○3万円~(月額)	○
TOTAL	◎	△?



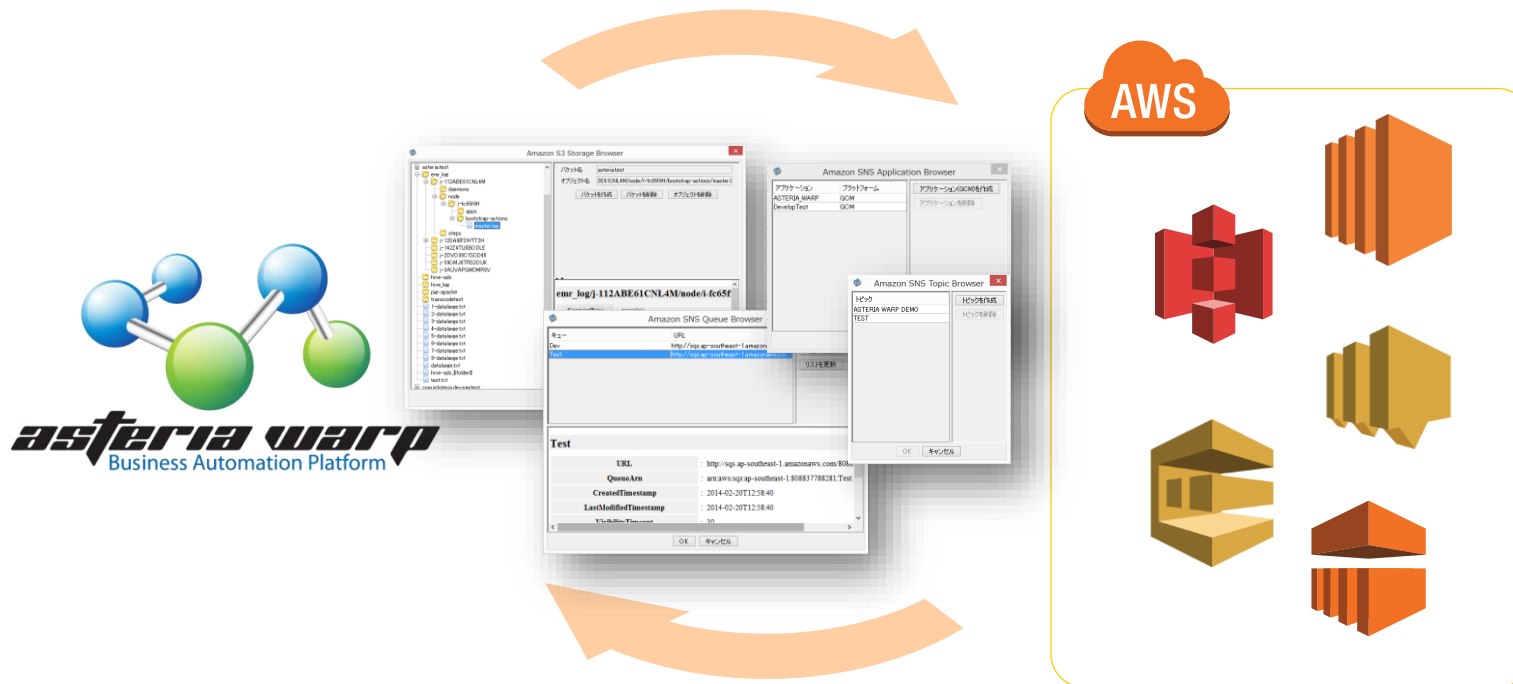


AWS 連携機能



Amazon Web Services アダプターオプション

- ASTERIA WarpからAWSで提供される様々なサービスに簡単な設定のみで接続することが可能なアダプター
- 専用のブラウザによって各サービスに接続、各種操作が可能





ASTERIA Warpが対応している AWSのサービス

■ ASTERIA WARPの標準機能で対応



■ Amazon RDS

Amazon RDSへの接続は通常のRDBと同様にSQLビルダーで操作



■ Amazon Redshift

ペタバイト級データウェアハウスのサービスであるAmazon RedshiftにもSQLビルダーで接続

■ AWSアダプターオプションで対応



■ Amazon S3

Amazon S3 へのデータ転送・取得をドラッグ&ドロップで構築
通常のストレージのように取り扱うことが可能



■ Amazon EC2

EC2インスタンスの起動終了を外部から制御
月次バッチ起動時のみインスタンスを起動するなどの運用が可能



■ Amazon DynamoDB

「DynamoDB Browser」の設定で、DynamoDB へのデータの入手力が可能



■ Amazon Simple Queue

専用画面の操作のみでキューを取り扱い



■ Amazon Elastic MapReduce

Elastic MapReduceを使ったAmazon環境でのHadoop処理を制御



■ Amazon Redshift

単純に読み取るだけであればSQLビルダーで可能だが、S3からのデータロードや分析に必要な機能を追加



■ Amazon SNS

ノーティフィケーションサービスを利用して、登録された端末にプッシュ通知を送信

コンポーネントの種類

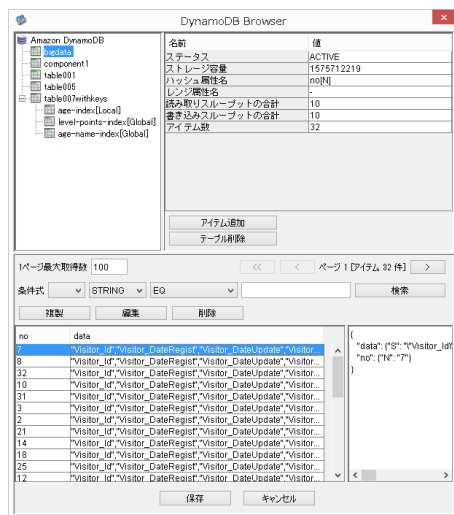
	コンポーネント名	機能概要
	AWS S3Get	Amazon Web ServicesからS3のコンテンツを取得します
	AWS S3Put	Amazon Web ServicesのS3へコンテンツを出力します
	AWS S3Download	Amazon Web ServicesのS3からファイル・フォルダーをダウンロードします
	AWS S3Upload	Amazon Web ServicesのS3へファイル・フォルダーをアップロードします
	AWS S3List	Amazon Web ServicesのS3からオブジェクトの一覧を取得します
	AWS S3Delete	Amazon Web ServicesのS3からオブジェクトを削除します
	AWS SQSGet	Amazon Web ServicesからSQSのメッセージを取得します
	AWS SQSPut	Amazon Web ServicesのSQSへメッセージを出力します
	AWS EC2SearchInstance	EC2のインスタンス情報を検索します
	AWS EC2GetInstance	EC2のインスタンスIDを指定してインスタンス情報を取得します
	AWS EC2CreateInstance	EC2のインスタンスを作成します
	AWS EC2CreateTag	EC2のインスタンスにタグ情報を設定します
	AWS EC2StartInstance	EC2のインスタンスを起動します
	AWS EC2StopInstance	EC2のインスタンスを停止します
	AWS EC2WaitStatus	EC2のインスタンスが指定されたステータスになるまで待機します
	AWS EC2TerminateInstance	EC2のインスタンスを終了します

	コンポーネント名	機能概要
	AWS EMRStart	Elastic MapReduceでジョブフローを開始します
	AWS EMRRunJob	Elastic MapReduceでジョブを開始します
	AWS EMRWaitJob	Elastic MapReduceで指定したジョブを待機します
	AWS EMRTerminate	Elastic MapReduceでジョブフローを停止します
	AWS SNSListTopic	SNSのトピック情報を取得します
	AWS SNSCreateTopic	SNSのトピック情報を取得します
	AWS SNSDeleteTopic	SNSのトピック情報を取得します
	AWS SNSListEndpoint	SNSのトピック情報を取得します
	AWS SNSCreateEndpoint	SNSのアプリケーションにエンドポイントを作成します
	AWS SNSDeleteEndpoint	SNSから登録されているエンドポイントを削除します
	AWS SNSListSubscription	SNSのトピックに登録されているサブスクリプション情報を取得します
	AWS SNSSubscribe	SNSのトピックにエンドポイントを登録します
	AWS SNSUnSubscribe	SNSから登録されていた受信登録を解除します
	AWS SNSPublish	SNSへメッセージを発行します
	AWS RedshiftLoad	S3からRedshiftへデータをロードします
	AWS RedshiftCopyTable	Redshiftに既存データのクエリーを元にしてテーブルを作成します



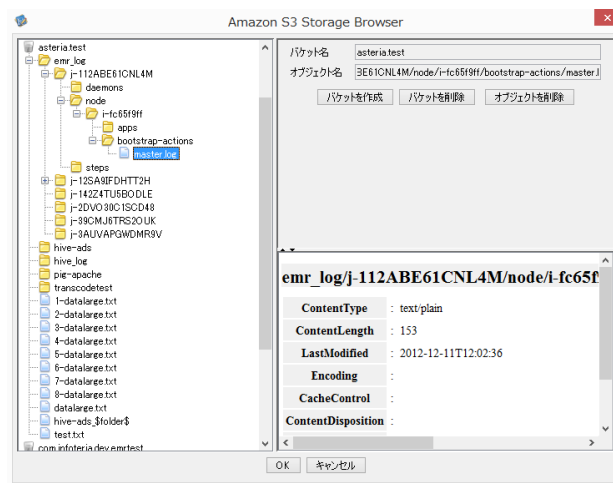
各種専用ブラウザ

■Amazon S3、DynamoDB、Simple Queue、SNS は専用の画面にて操作可能



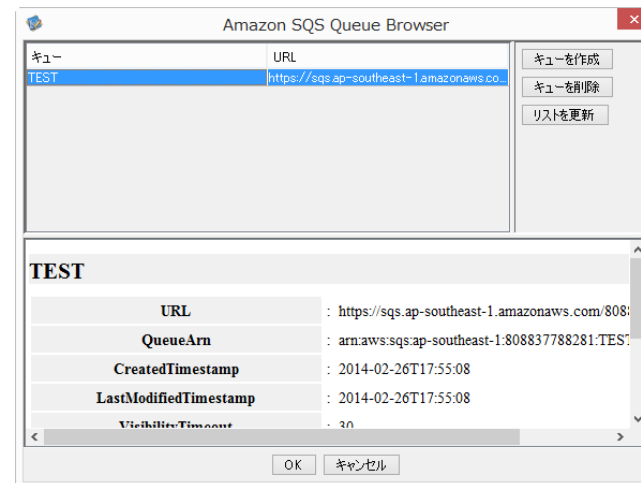
■Amazon DynamoDB Browser

接続したDynamoDBのテーブル、インデックスをツリー構造で表示し、操作



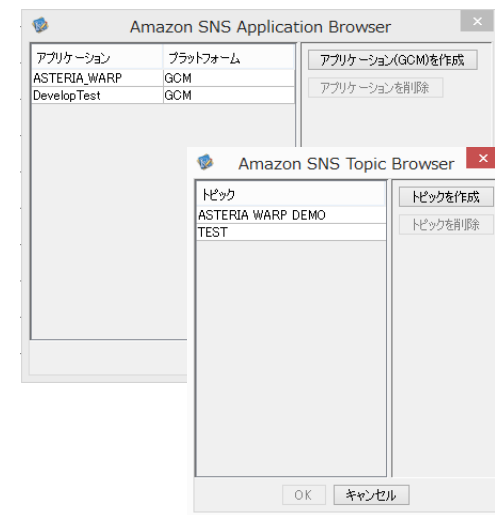
■Amazon S3 Storage Browser

通常のストレージのようにツリービューを持った直感的な画面で操作



■Amazon SQS Queue Browser

キューの一覧と作成削除を直感的な画面にて操作



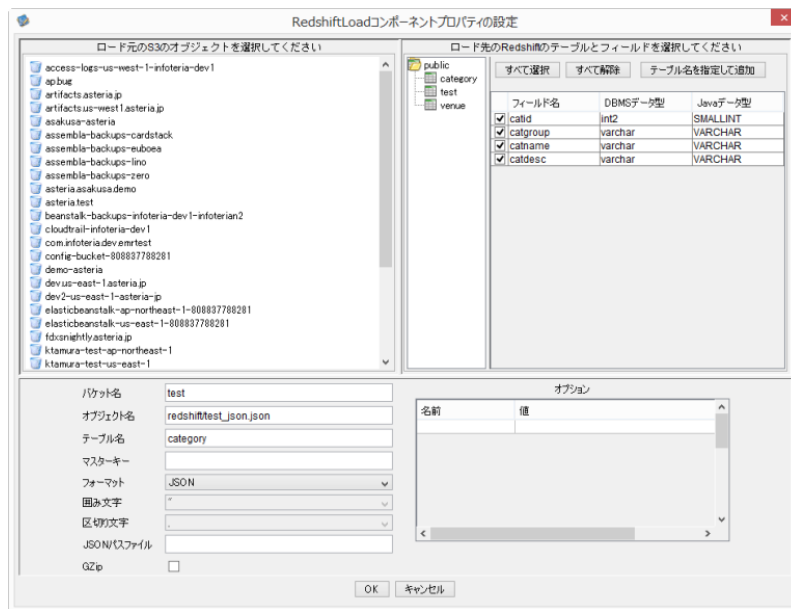
■Amazon SNS Topic Browser, Application Browser

アプリケーション、トピックを一覧表示、操作を行う画面を用意



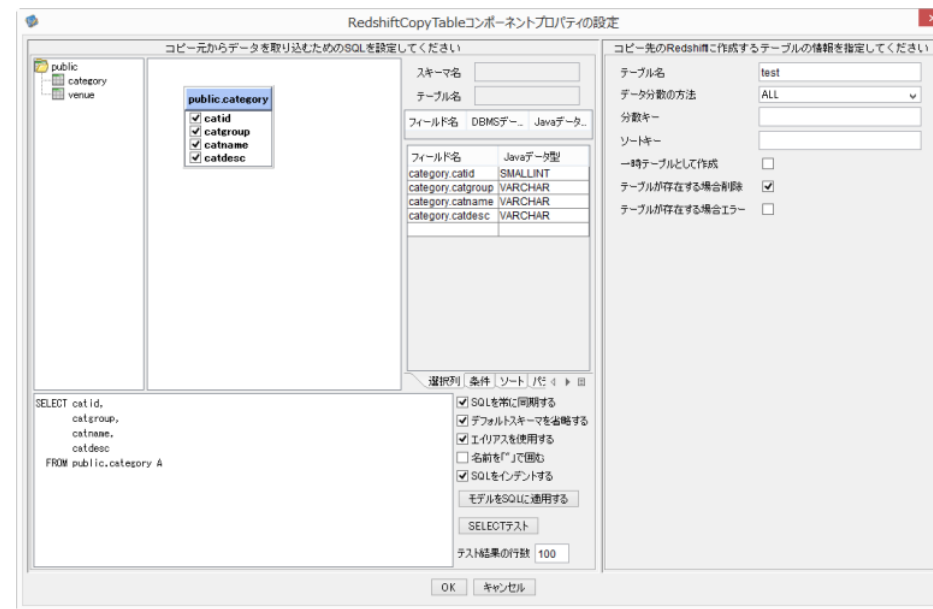
Redshift専用ブラウザ

- Amazon Redshiftを使った分析業務では以下の2つのコンポーネントで支援



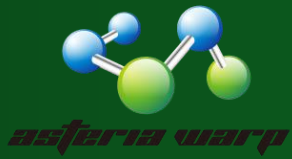
AWS RedshiftLoad

S3からRedshiftへのデータロードを画面の選択のみで実現



AWS RedshiftCopyTable

分析時に必要となる一時テーブルの作成も直感的な画面操作で実現



AWS 連携パターンとDemo



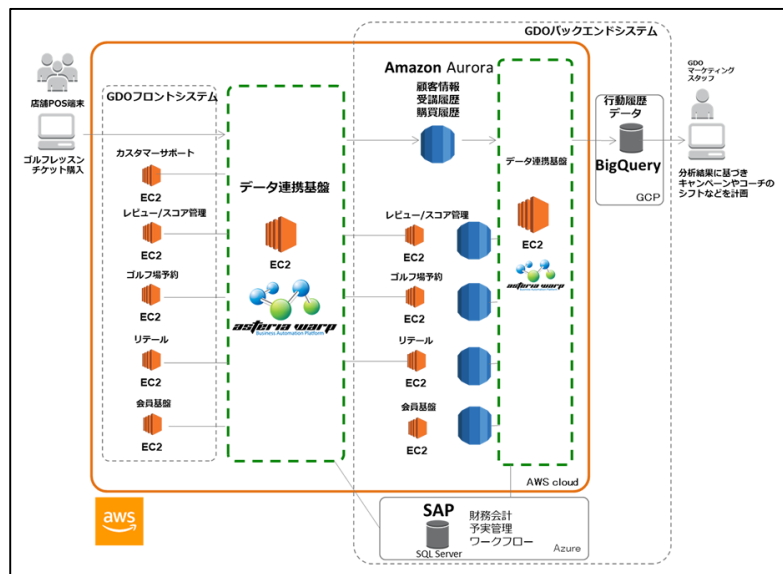


Amazon RDS(Aurora)へのデータ入出力



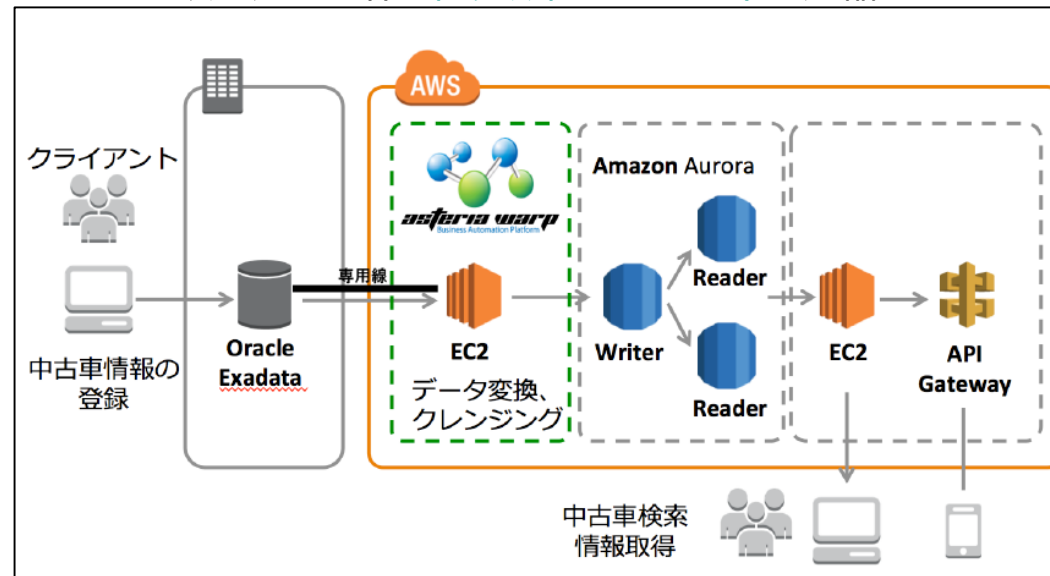
事例1: ゴルフダイジェストオンライン様

ECサイトからバックエンドシステムまで社内主要システムを疎結合で連携するデータ連携基盤として利用している。
ASTERIA Warp処理件数は現在では1日平均400万件以上に及ぶ。



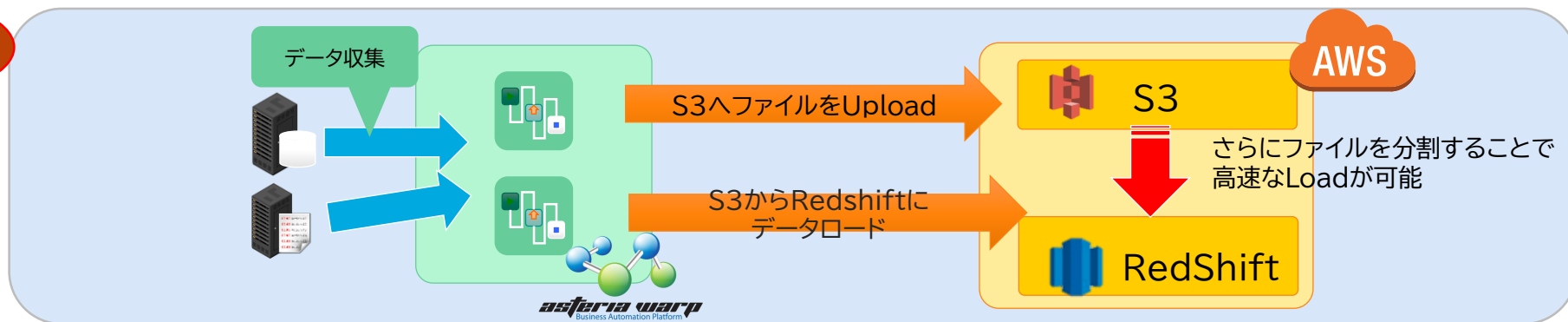
事例2: プロトコーポレーション様

ASTERIA Warpの並列処理機能により8スレッド同時書き込みを行うことで、性能を高め5分おきの確実な差分更新を実現。
項目追加に伴う開発期間を1/5以下に短縮。



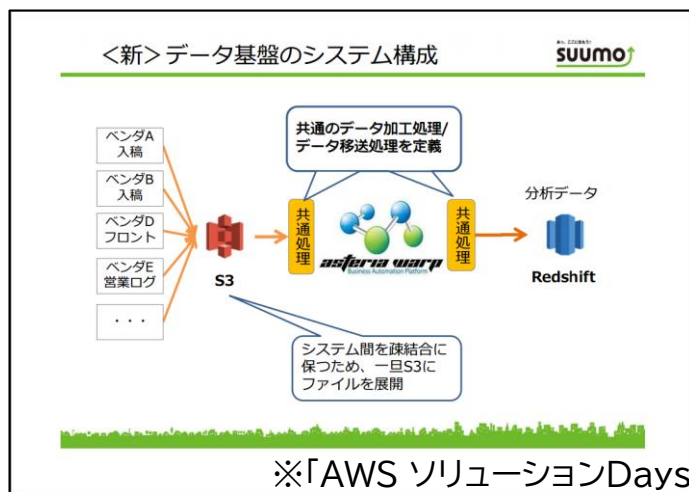
Amazon RedShiftへのデータ登録

Demo



事例1:リクルート住まいカンパニー様

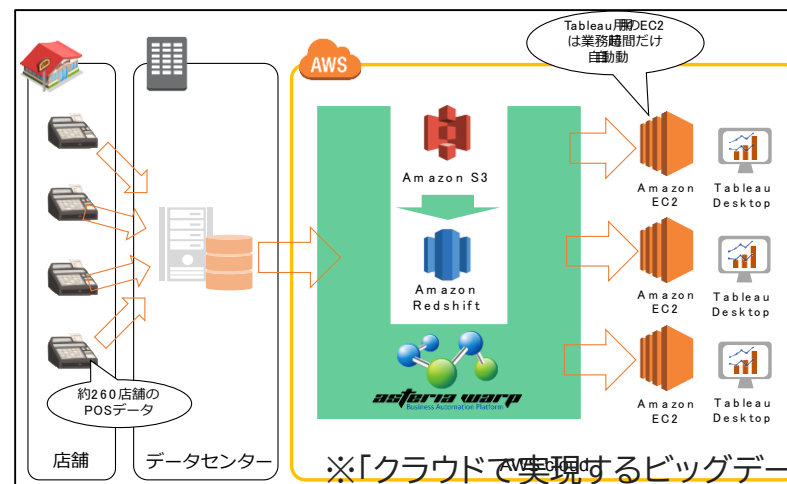
膨大なデータの蓄積基盤としてS3/Redshiftを、データの移送処理、加工処理の共通化にASTERIAWARPを採用
4か月で数百バッチの置き換え(+共通化)を完了、計画の1.5倍の進捗



※「AWS ソリューションDays2」講演資料より抜粋

事例2:イオンイーハート様

POSデータ分析基盤をRedshift × Tableauで構築
気象情報を始めとした先進的な統合分析を日々刷新中



※「クラウドで実現するビッグデータ分析最前線」にて登壇

Amazon Rekognition の利用



ASTERIA Warp 機能名		説明
	AWS RekognitionOCR	画像から文字列を取得
	AWS RekognitionTagImage	画像にタグ付け
	AWS RekognitionImageModeration	画像のモデレーション

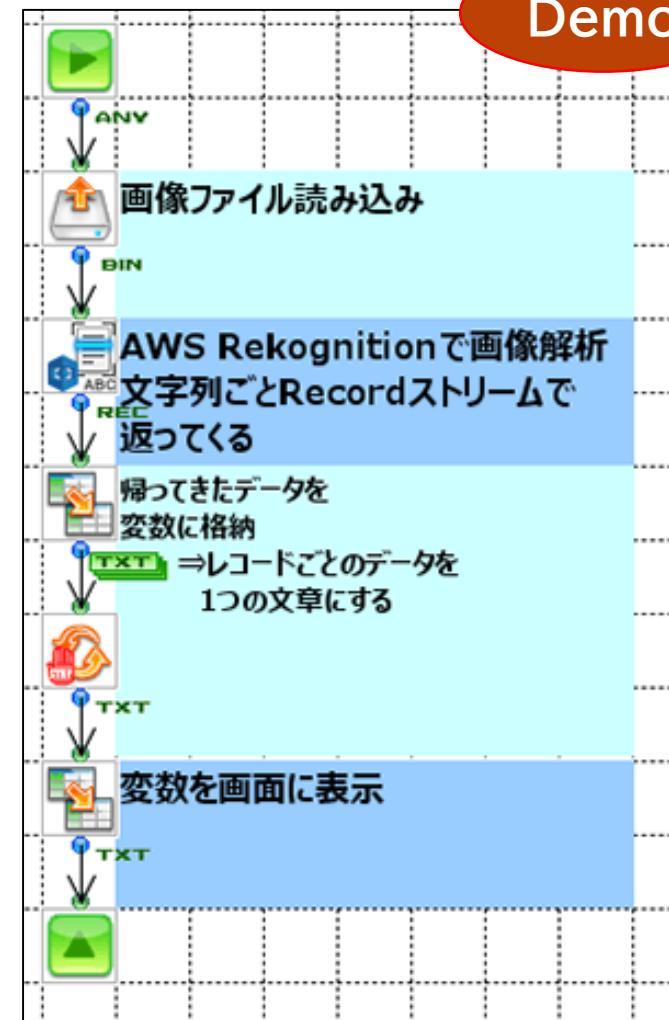


例えば・・・

- 伝票スキャンデータのOCR処理
- 不適切なコンテンツ・テキスト投稿の監視

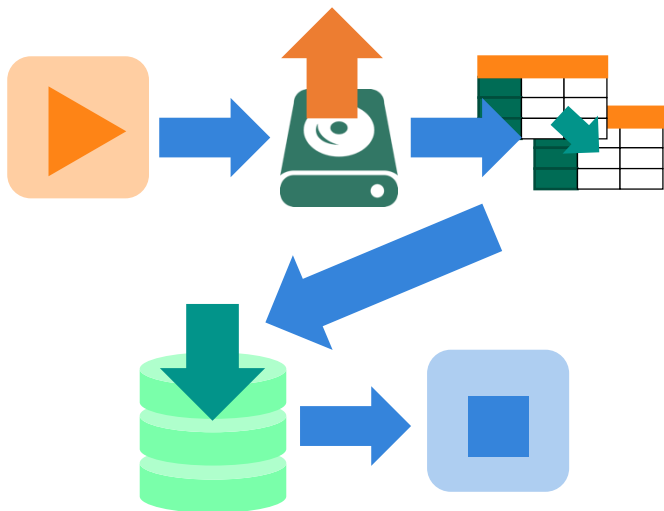
Experimental Buildで公開中！

Demo



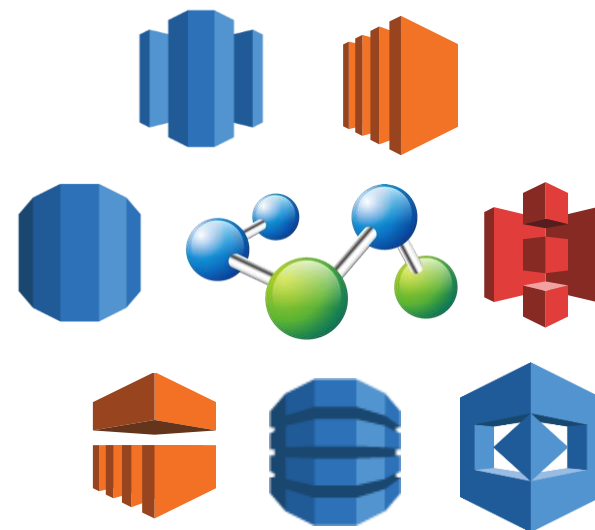


まとめ

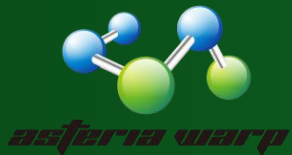


プログラミングやAPIの知識を必要とせず、
月額3万円から利用可能

基盤利用で多数の実績と、
豊富なAWS連携機能を持つ



お試しください！



お知らせ



今だけ！無料体験のお申込みで”全員”に使い方書籍をプレゼント！
「手ぶら de ASTERIA Warp」で 業務自動化体験



- ✓ 5日間無料で体験
- ✓ サーバー準備不要
- ✓ スタートガイド付き

手ぶらでアステリア



Asteria[✦]

Thank you.