

スマラジ SmartDB Radio

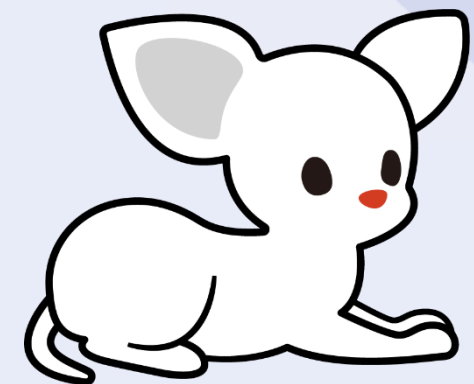
もっと活用！

「SmartDB」Q&A 会

学ぶ！ SmartDB×他システム連携の第一歩 vol.2
～データ連携を体験してみよう【プチハンズオン】～



2022年12月7日
株式会社ドリーム・アーツ





本日の講師紹介：

足立 翔太郎 (Shotaro Adachi)

サービス&プロダクト開発本部
SmartDBグループエンハンスチーム

2022年10月にドリーム・アーツに入社
11月末までカスタマーサクセス部門での
製品のお問い合わせ対応業務を経て、
現在は製品開発に従事

ドリーム・アーツ入社前は、オーストラリアにて
バリスタをやっていました。

ユーザの皆さんが使いやすい製品
を目指して頑張ってます！



“スマラジ！”とは SmartDB Radio



SmartDBコミュニティの紹介



特定テーマ 議論の会



特定機能 深堀りの会

SmartDBの定期開催イベント

今までのスマラジ！動画を公開



※閲覧にはサポートサイトへのサインインが必要です。

SmartDBコミュニティサイトにて、
8月10日（水）開催：スマラジ！学ぶ！SmartDB×他システム連携の第一歩
スマラジ！講演動画&資料を公開しています！

本日のテーマは…

SmartDB APIを使ったハンズオン



本日の流れ

No.	時間	所要	内容
1	15:55	5分	お願い事項、企画趣旨について
2	16:00	45分	API概要、SmartDB APIについて PowerShellを使ったプチハンズオン 外部連携ツールの役割について
3	16:45	10分	質疑・応答
4	16:55	5分	お知らせ
5	17:00		終了（アンケートのお願い）

★終了後30分ほど講師は残ります。ご質問などにご活用ください。

0. 本日のゴール



- SmartDB API を利用した業務のイメージを掴む
 - SmartDB API を利用する目的に対して、必要となる準備を理解する

皆さんの飲み物のマイブーム教えてください



1. SmartDB APIとは



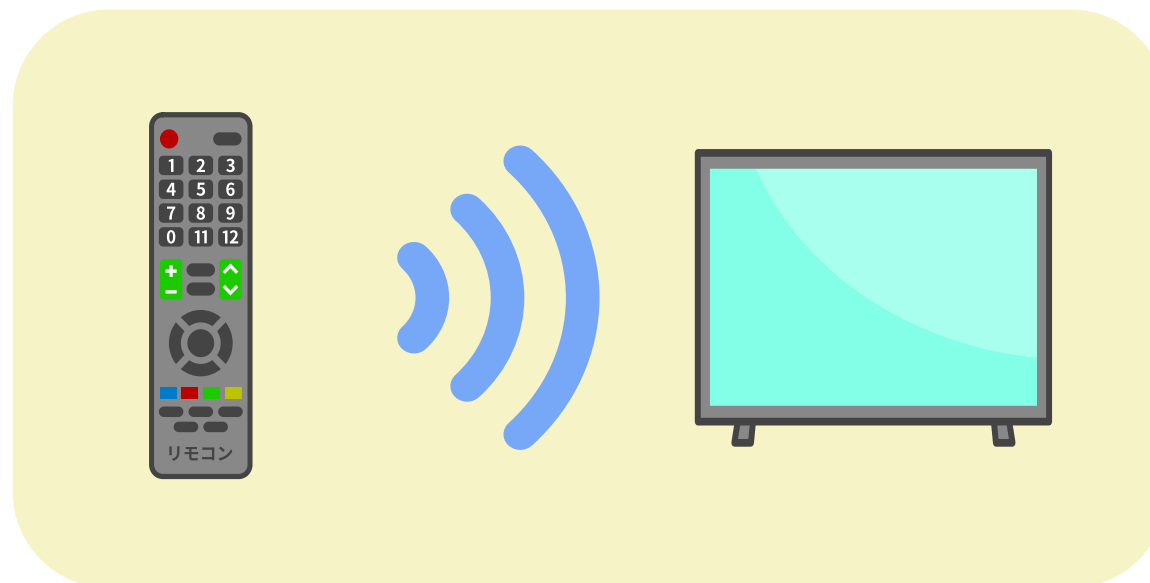
API (Application Programming Interface)

プログラムからアプリケーションを操作するために
アプリケーション側が提供する機能

APIとは

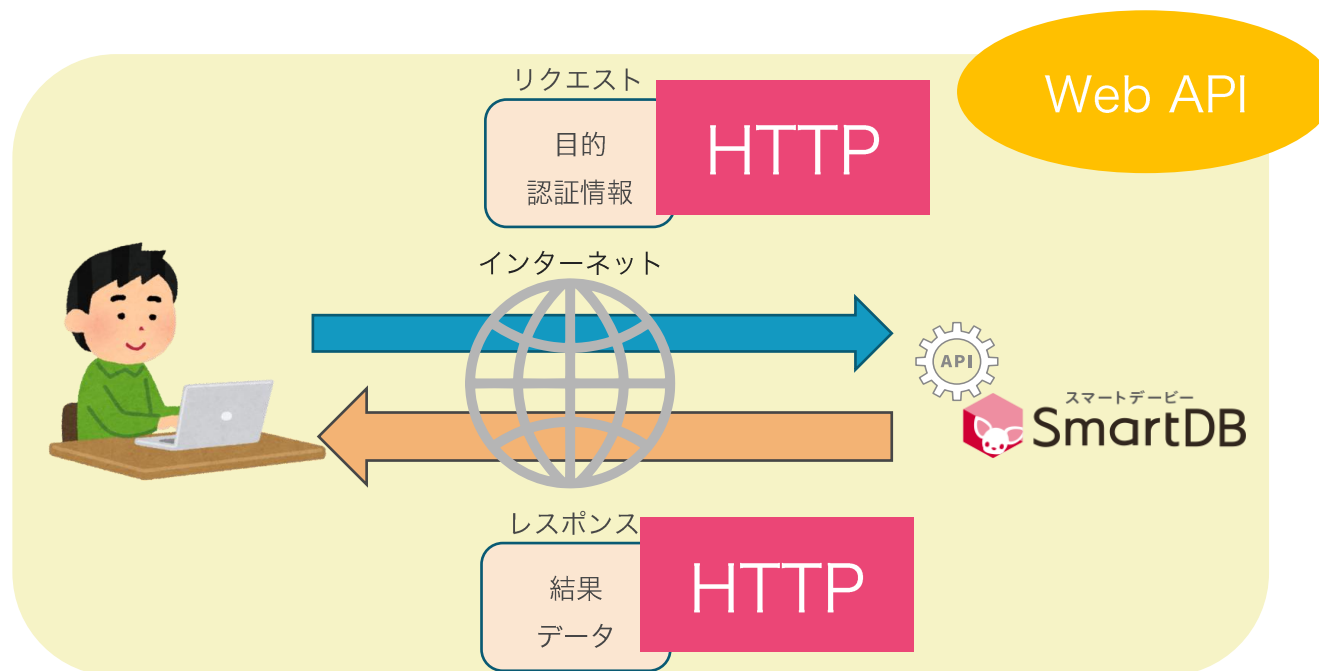
例：テレビのリモコン

- リモコンからテレビに無線・赤外線を送る
 - ・ チャンネルを変更したい
 - ・ 音量を上げたい
- テレビは信号に応じて設定を変更する



Web APIとは (例: SmartDB)

- ユーザは SmartDB API にリクエストを送る
 - 新規文書をつくりたい
 - 文書一覧を取得したい
- SmartDB APIはリクエストに応じてデータ更新を行い、結果・データを返す
- リクエストには認証が必要



Web APIとは

APIリクエスト



DreamArts

その他のサービス



顧客管理システム
(CRM)

社内システム

SmartDB の提供するサービス

好 好きなドリンク ...

文書登録

一覧 下書き保存 登録

好きなドリンク

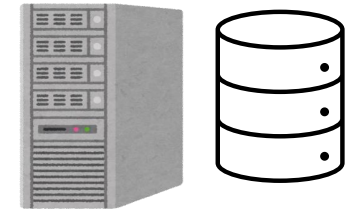
氏名	<input type="text"/>
カテゴリ	コーヒー
お店・銘柄	<input type="text"/>

SmartDB Web ページ

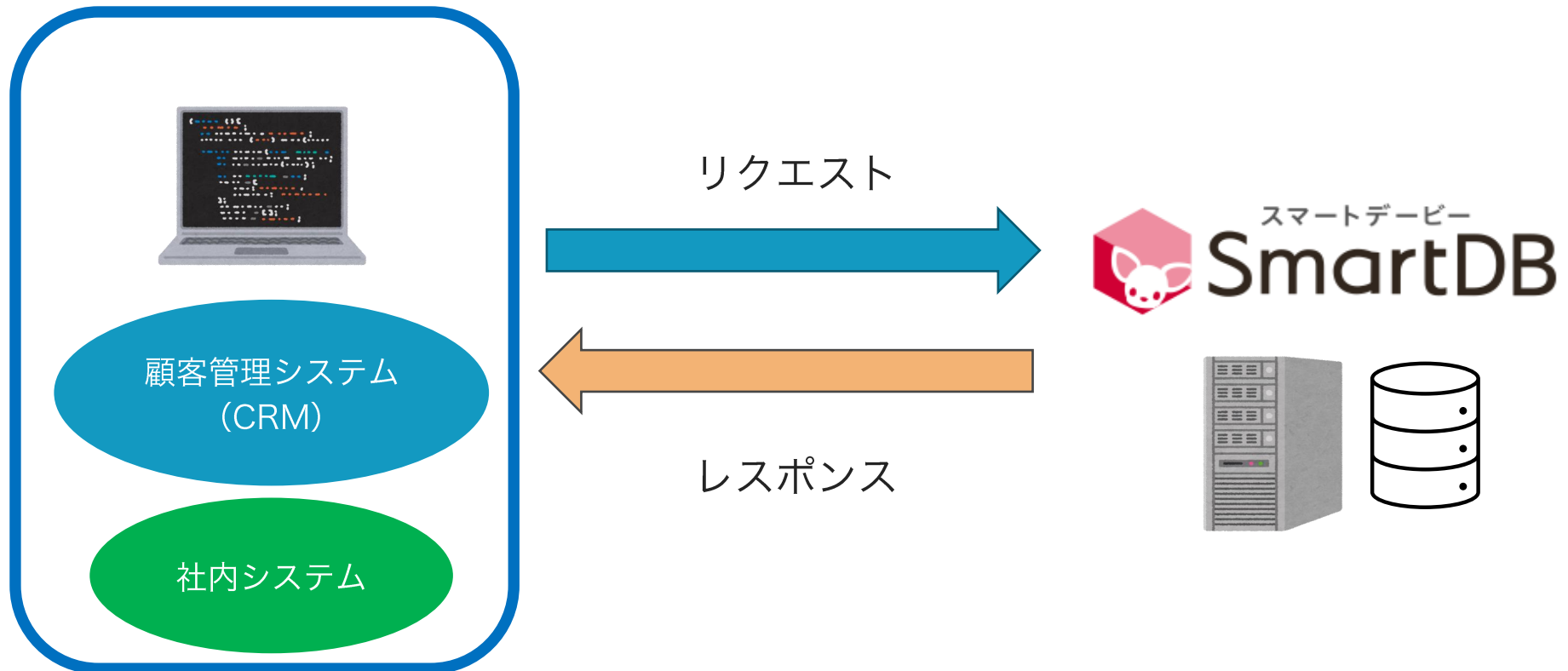


スマートデービー

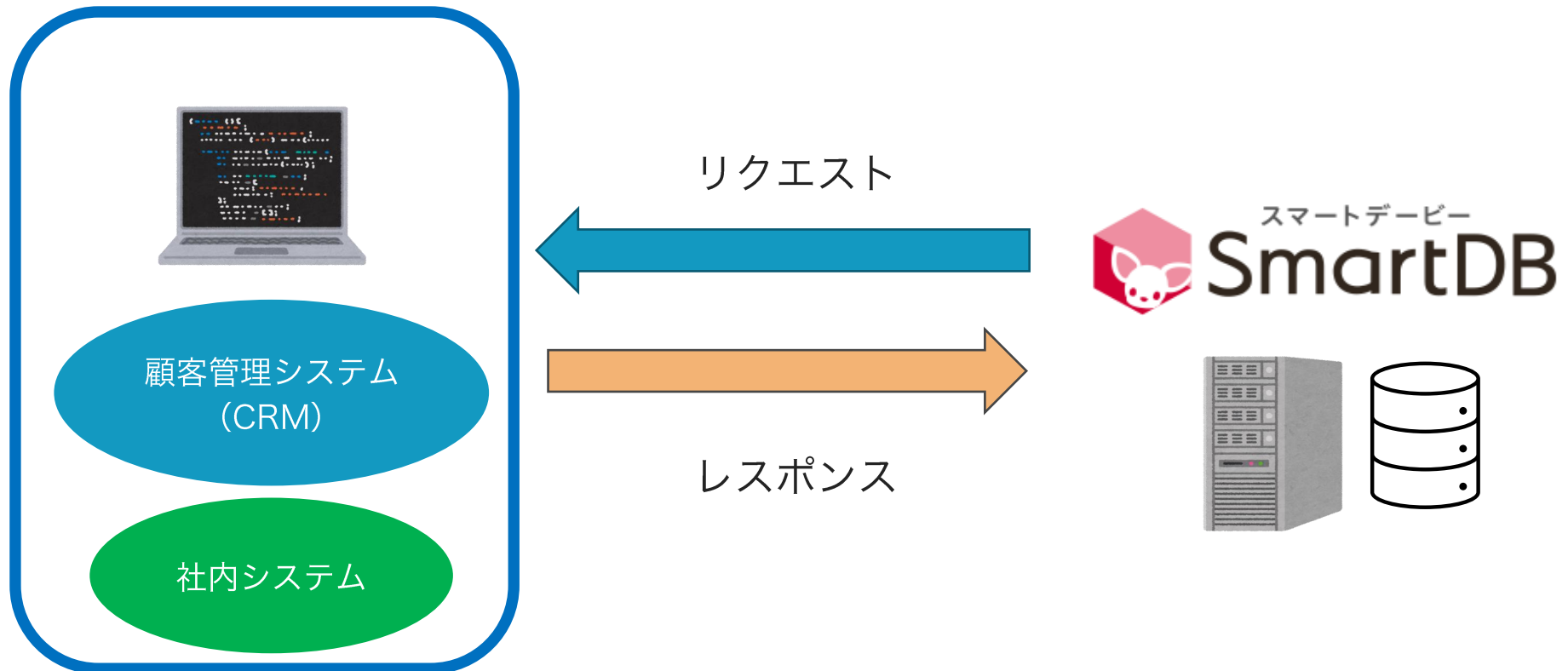
SmartDB



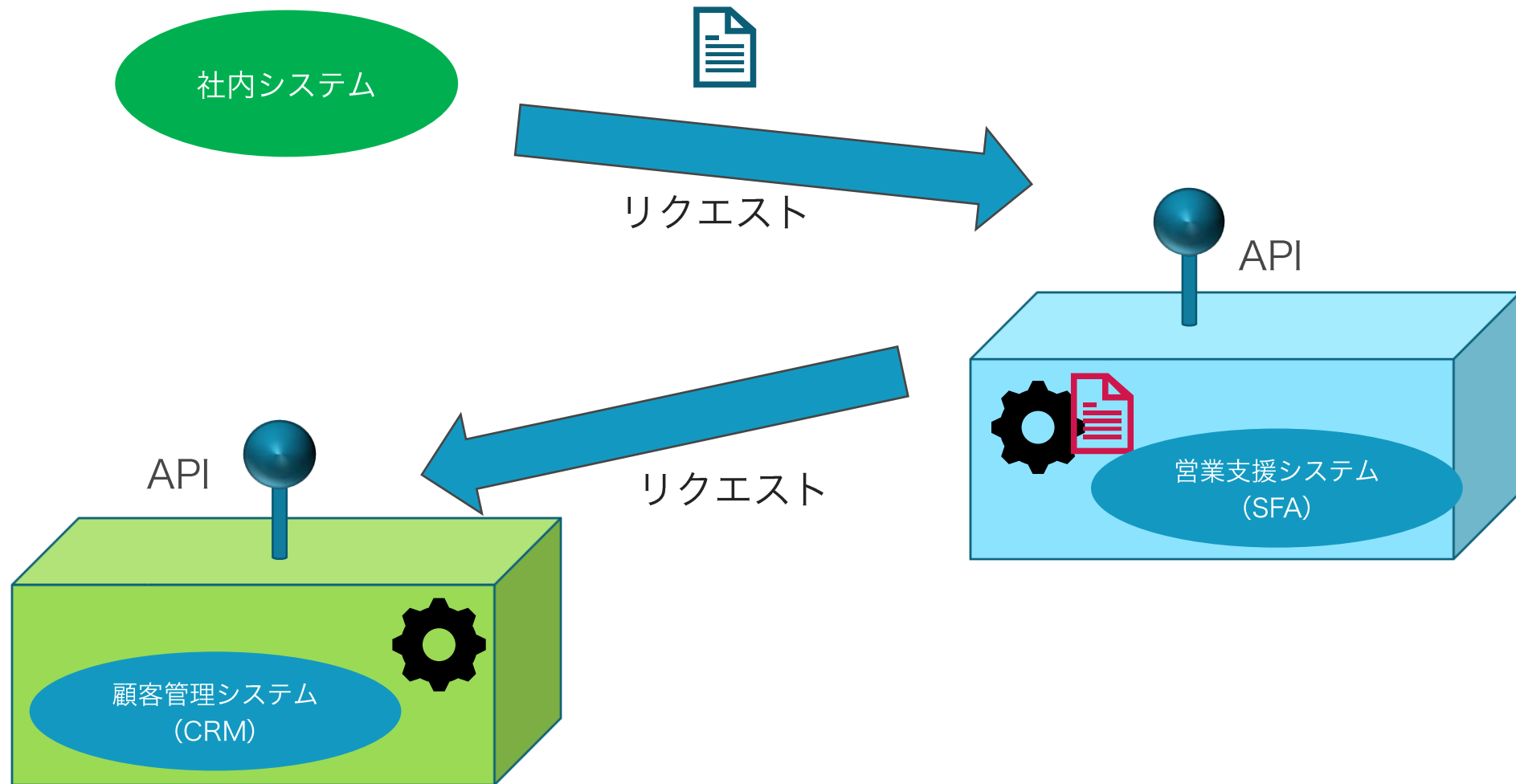
Web APIとは



Web APIとは



理想は . . .

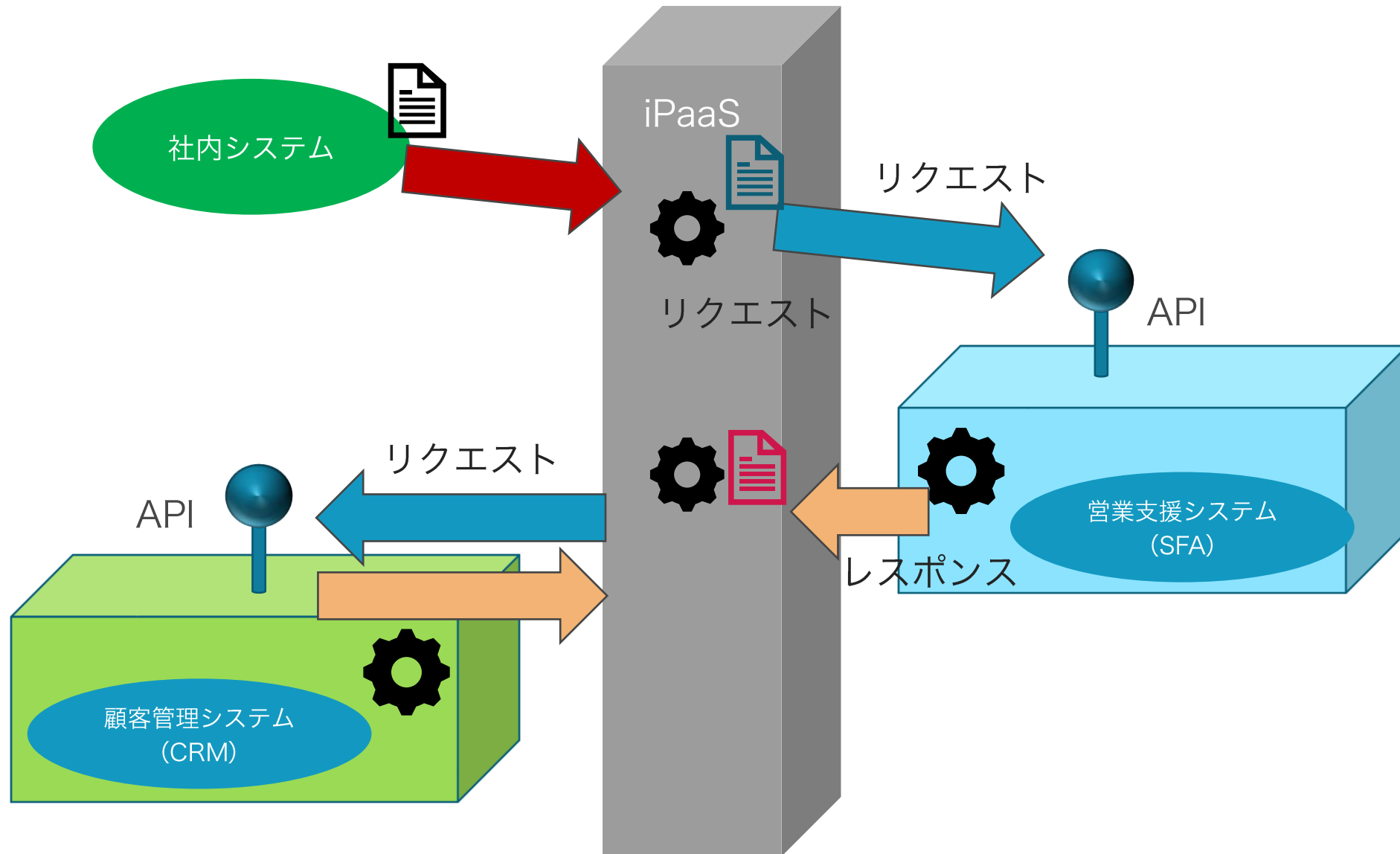


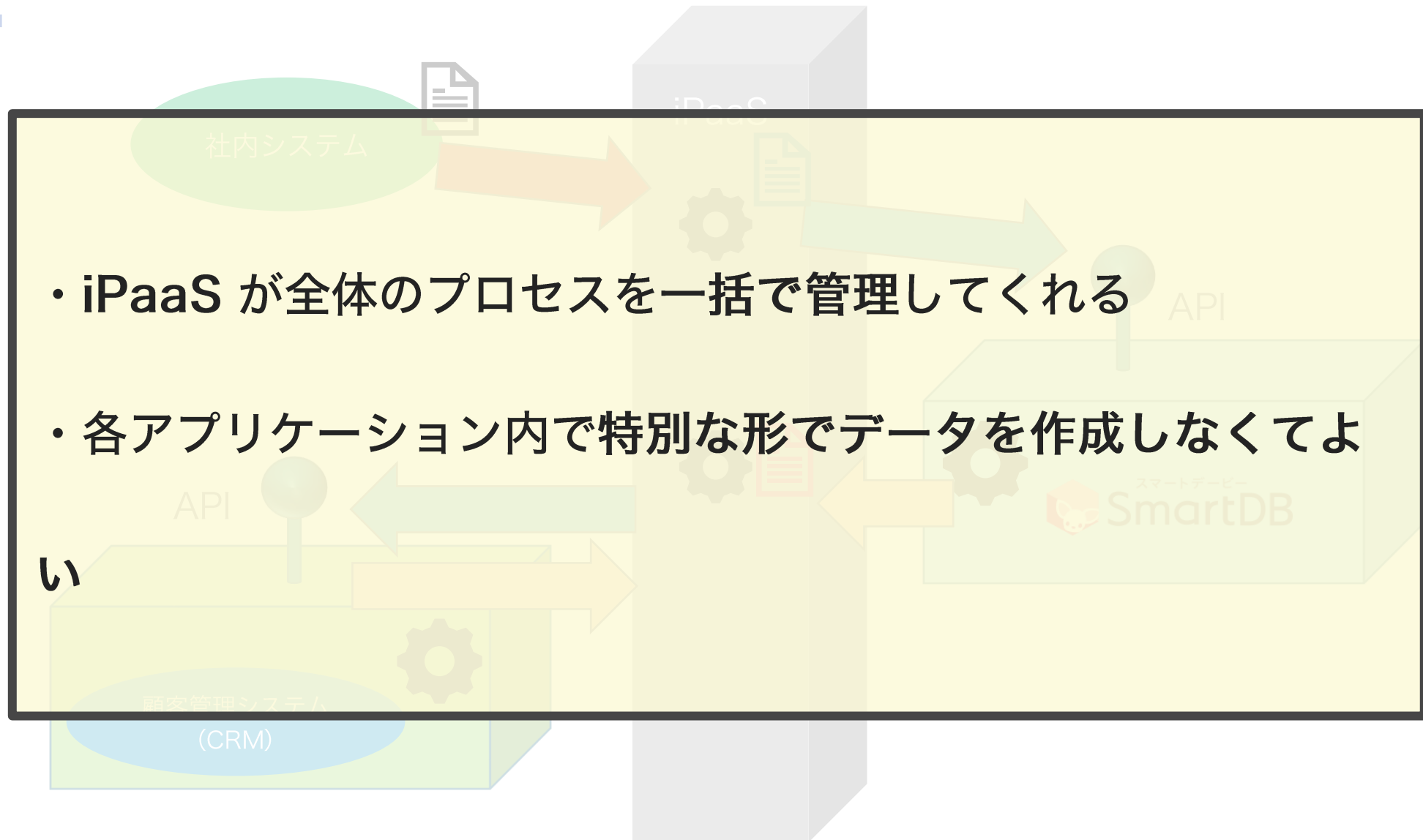
-
- 各アプリケーションから全体のプロセスが管理しにくい
 - 次のアプリケーションが受け入れることのできる形のデータをアプリケーション内で作成しなければならない
 - そもそも実現が困難な場合が多い

iPaaS (Integration Platform as a Service)

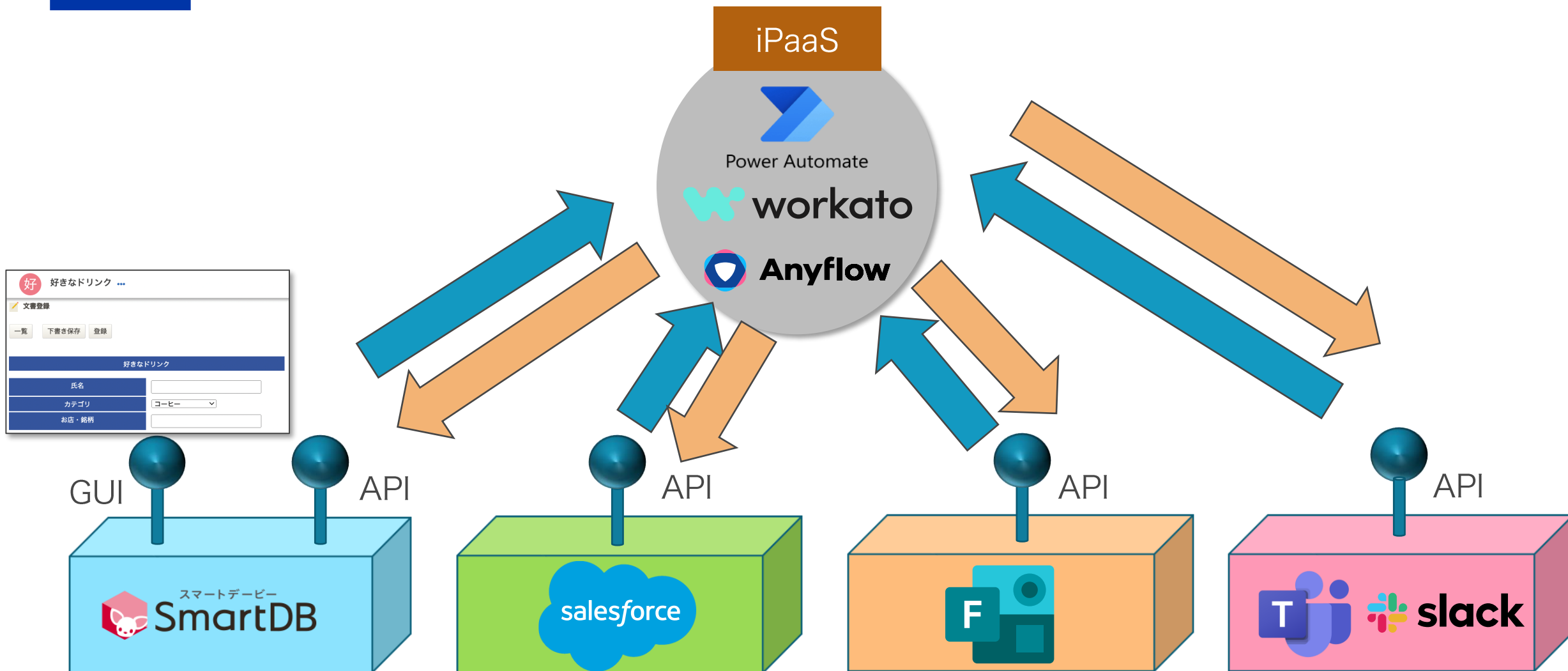
複数のサービス・アプリケーションを利用した
統合プロセスの開発・実行を行うサービス

iPaaS とは



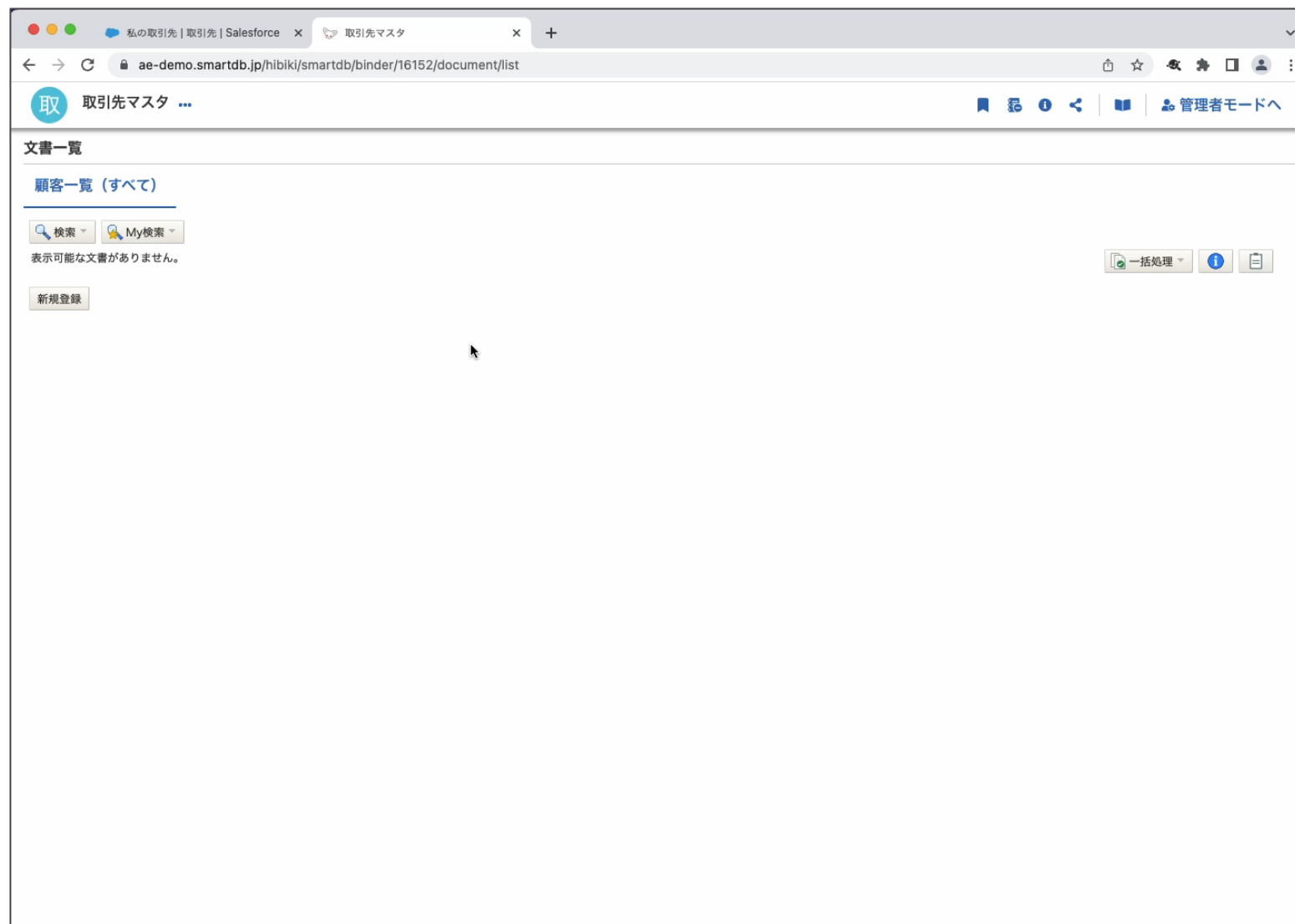


Web API利用のイメージ

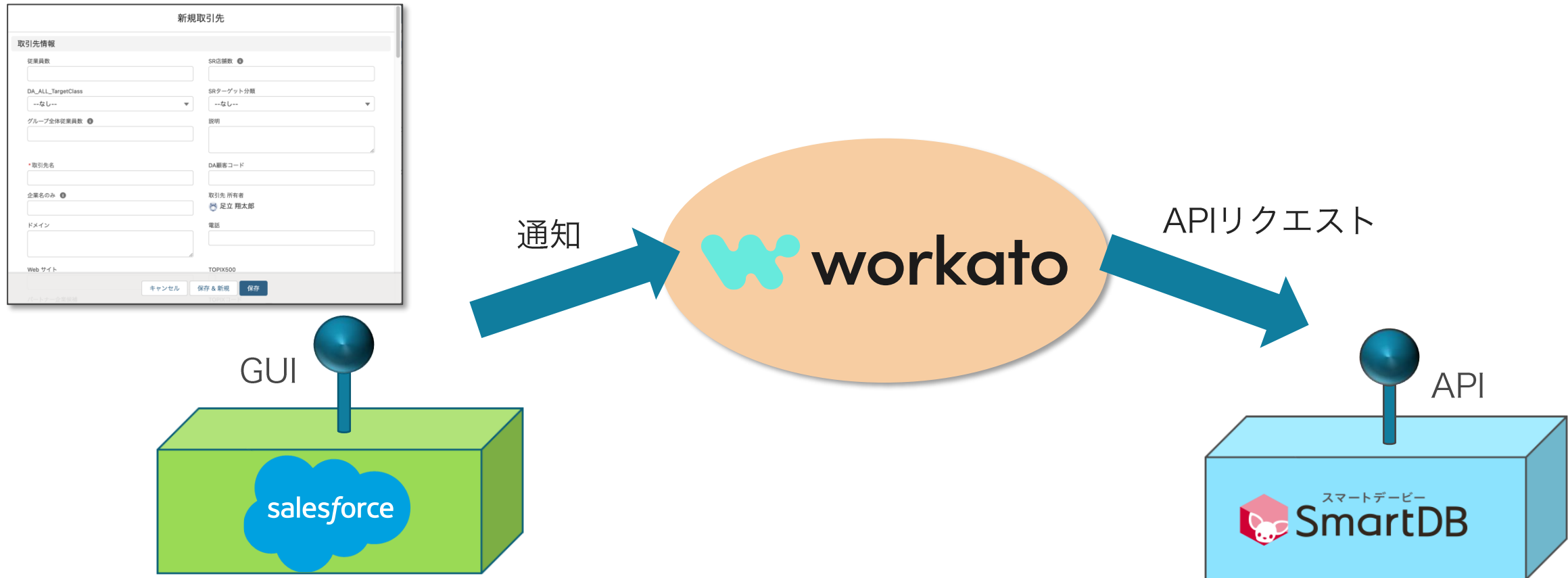


- Microsoft Forms へ回答が入力される
→ Microsoft Power Automate を仲介して SmartDB の新規文書として登録する
- Salesforce などの顧客管理システム（CRM）に新規の取引先データを登録
→ Workato を仲介して SmartDB にも同期する
- Microsoft Teams や Slack などのチャットボットへデータを送信
→ Workatoを仲介して SmartDB への文書登録

Web API連携の例



Web API連携の仕組み

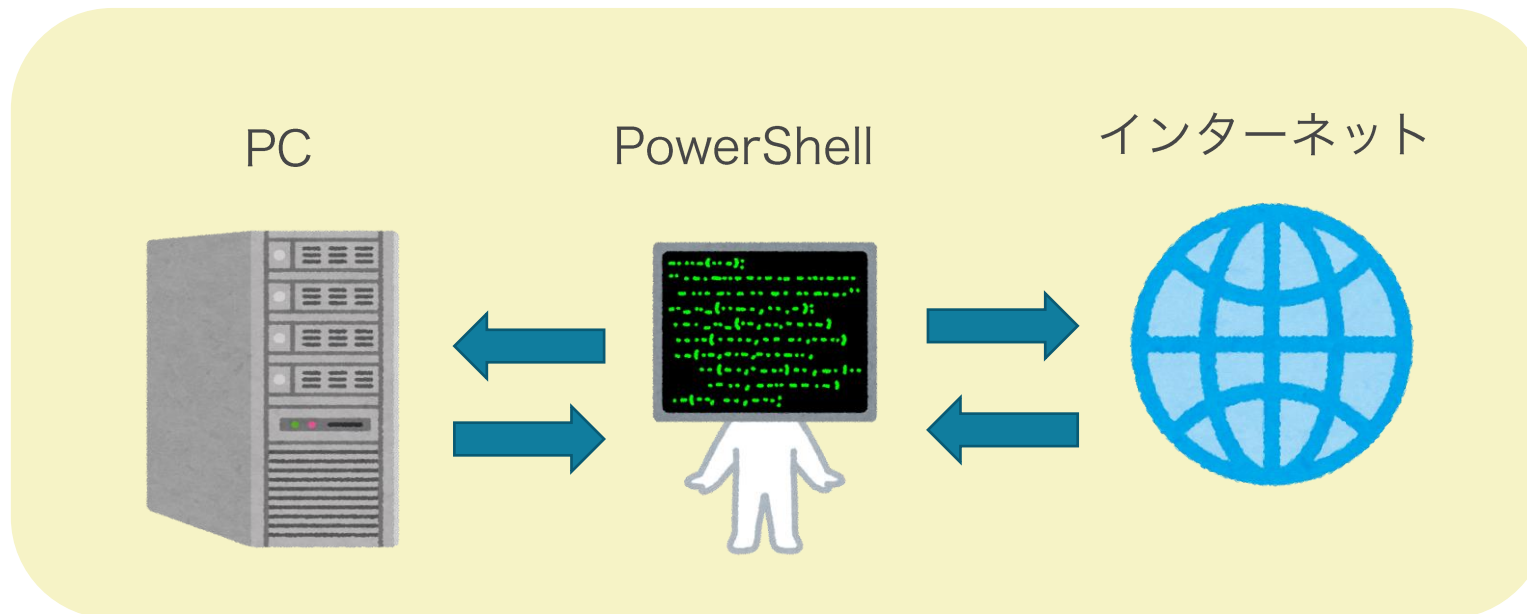


2. PowerShellを使った プチハンズオン



- PowerShellから SmartDB に APIリクエストを送信するイメージをつかむ
- どのような業務に対して、直接リクエストを作成して送信する方法が有効となるかを見極められるようになる

- Windowsに標準搭載されているコマンドラインツール
 - PCの動作制御（ファイル操作・アプリケーションの起動）
 - HTTPリクエストの送信

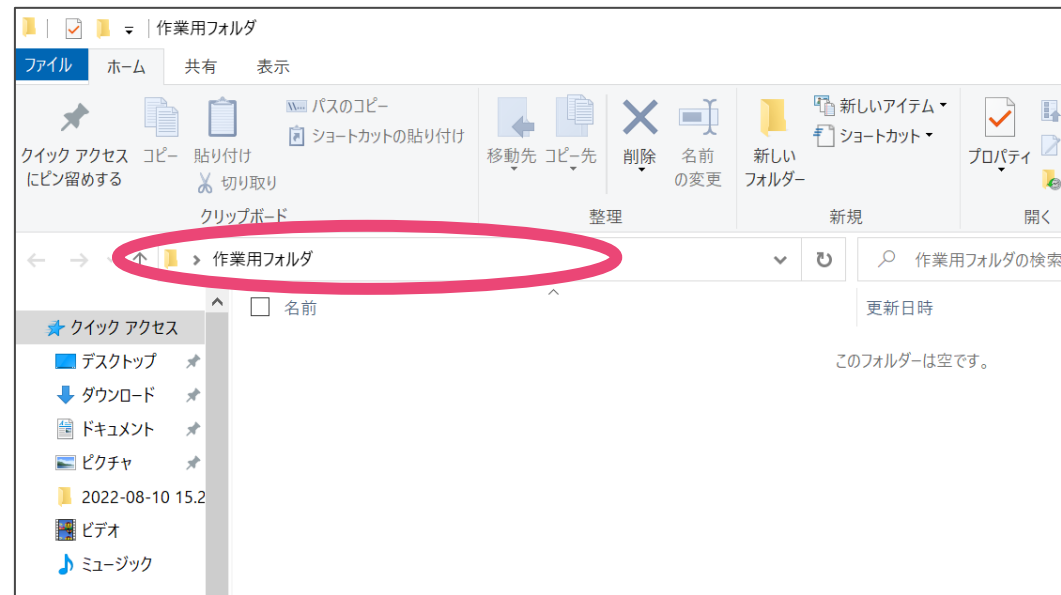


1. 作業フォルダの作成

任意の場所（例えばデスクトップ上）に新規フォルダを作成してください

2. PowerShell の起動

1. で作成したフォルダ内のアドレスバーをクリックし、
「powershell」と入力し、エンターキーを押す



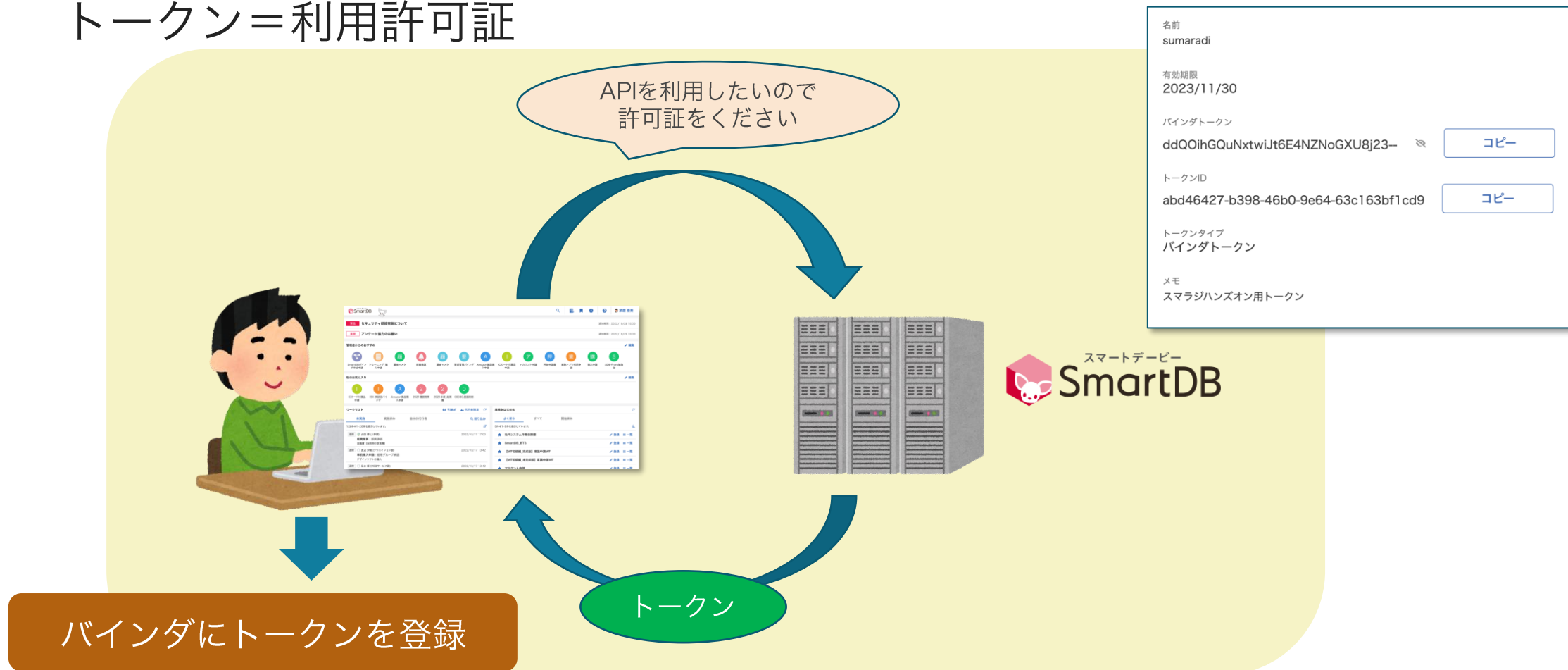


- メモ帳の起動

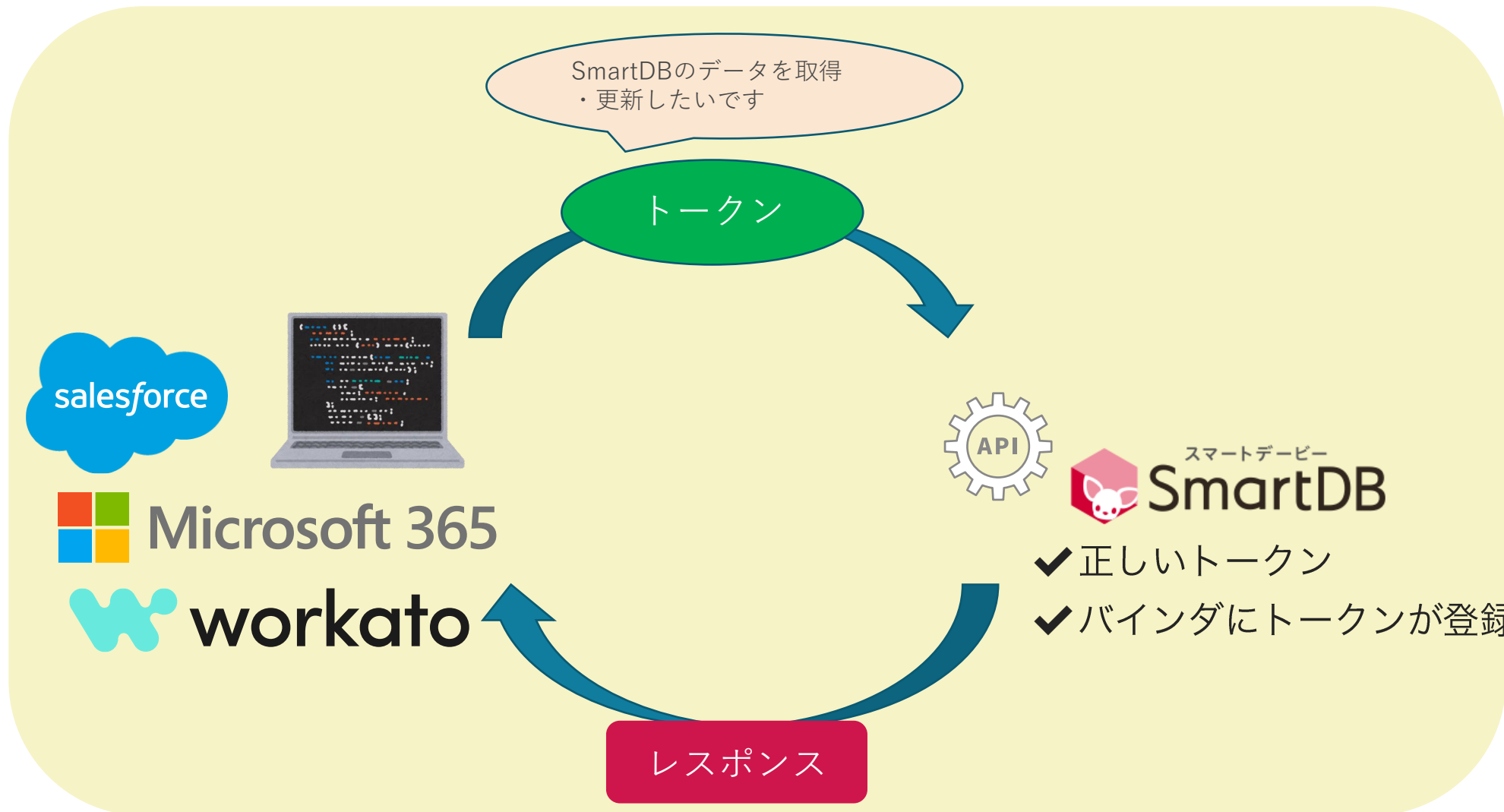
```
notepad
```


SmartDB API へのリクエスト

- トークン認証 = SmartDB APIの認証方法
トークン = 利用許可証



SmartDB API へのリクエスト



- 今回使用するバイндаの構成

部品キー	部品名	部品種別
Name	氏名	テキスト
Category	飲み物の種類	単一選択
Brand	飲み物の銘柄・店	テキスト

① 文書一覧の取得

```
$req1 = @{
    URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/views/10001/documents"
    Method = "GET"
    Headers = @{
        Authorization = "Bearer tLW6IXW0mqrIh2MfDLW0wtWVixkRcTzC"
    }
}
Invoke-WebRequest @req1 -UseBasicParsing -OutFile .¥response1.json
$res1 = ((Get-Content .¥response1.json -Encoding UTF8) | ConvertFrom-Json).documents.docum
echo $res1
```

1

```
Windows PowerShell
PS C:\Users\志水泰貴\OneDrive - ドリーム・アーツ\デスクトップ\work>
PS C:\Users\志水泰貴\OneDrive - ドリーム・アーツ\デスクトップ\work>
PS C:\Users\志水泰貴\OneDrive - ドリーム・アーツ\デスクトップ\work>

*無題 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
$req1 = @{
    URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/views/10001/documents"
    Method = "GET"
    Headers = @{
        Authorization = "Bearer tLW6lXW0mqrIh2MfDLW0wtWVixkRcTzC"
    }
}
Invoke-WebRequest @req1 -UseBasicParsing -OutFile .\response1.json
$res1 = ((Get-Content .\response1.json -Encoding UTF8) | ConvertFrom-Json).documents.document
echo $res1

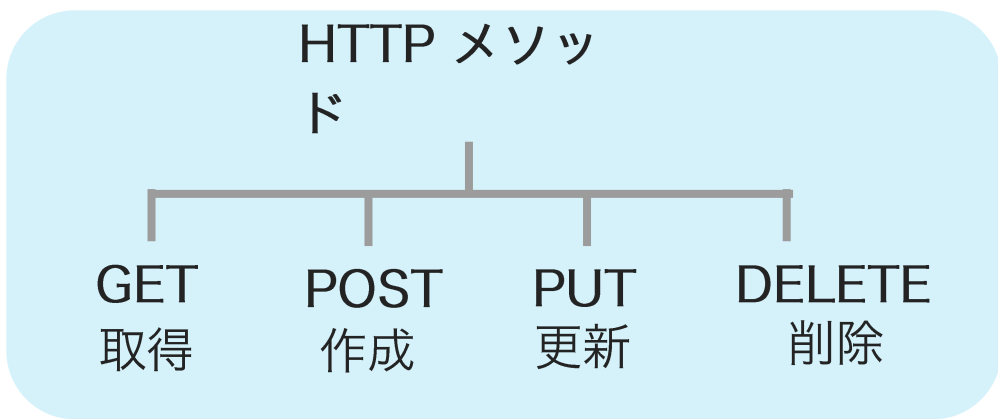
11行、1列 100% Windows (CRLF) UTF-8
```

① 文書一覧の取得

```
$req1 = @{  
  URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/views/10001/documents"  
  Method = "GET"  
  Headers = @{  
    Authorization = "Bearer tLW6lXW0mqrIh2MfDLW0wtWVixkRcTzC"  
  }  
}
```

バイндаID

ビューID



②-1 文書新規登録

```
$req2_1 = @{
  URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/documents"
  Method = "POST"
  Headers = @{
    Authorization = "Bearer tLW6IXW0mqrh2MfDLW0wtWVixkRcTzC"
    ContentType = "application/json"
  }
  Body = @{
    "Name" = "足立 翔太郎"
    "Category" = 10
    "Brand" = "ゲイシャ"
  }
}
Invoke-WebRequest @req2_1 -UseBasicParsing -OutFile .¥response2_1.json
$res2_1 = ((Get-Content .¥response2_1.json -Encoding UTF8) | ConvertFrom-Json).document
echo $res2_1
```

半角・「"」なし

10 : コーヒー
11 : お茶
12 : 水
13 : ジュース
14 : アルコール類

②-2 文書一覧の取得

```
$req2_2 = @{
  URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/views/10001/documents"
  Method = "GET"
  Headers = @{
    Authorization = "Bearer tLW6IXW0mqrIh2MfDLWowtWVixkRcTzC"
  }
}
Invoke-WebRequest @req2_2 -UseBasicParsing -OutFile .\response2_2.json
$res2_2 = ((Get-Content .\response2_2.json -Encoding UTF8) | ConvertFrom-Json).documents.docu
echo $res2_2
```

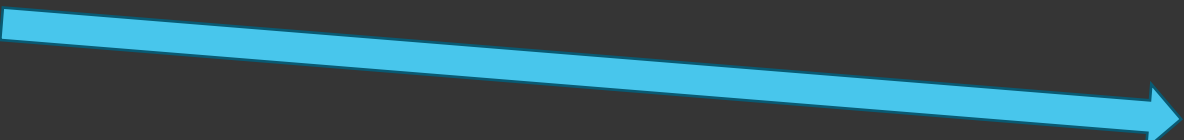
③ 文書削除

```
$req3 = @{  
  URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/documents/10"  
  Method = "DELETE"  
  Headers = @{  
    Authorization = "Bearer tLW6IXW0mqrIh2MfDLW0wtWVixkRcTzC"  
  }  
}  
Invoke-WebRequest @req3 -UseBasicParsing -OutFile .\response3.json  
$res3 = ((Get-Content .\response3.json -Encoding UTF8) | ConvertFrom-Json).document  
echo $res3_2
```

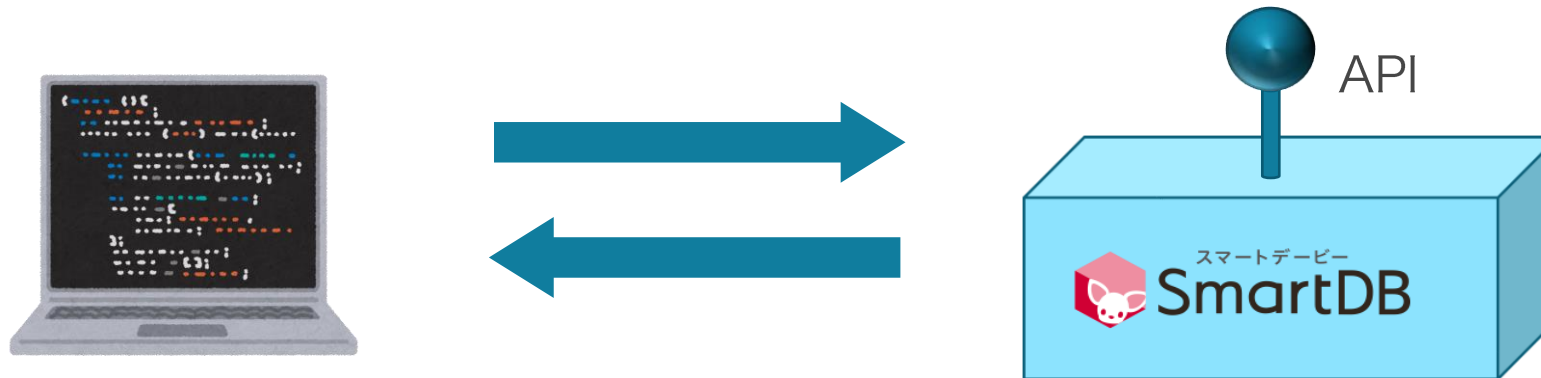
文書番号

③ 文書削除

```
$id = $res2_1.id
$req3 = @{
  URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/documents/${id}"
  Method = "DELETE"
  Headers = @{
    Authorization = "Bearer tLW6IXWOmqrIh2MfDLWowntWVixkRcTzC"
  }
}
Invoke-WebRequest @req3 -UseBasicParsing -OutFile .\response3.json
$res3 = ((Get-Content .\response3.json -Encoding UTF8) | ConvertFrom-Json).document
echo $res3
```

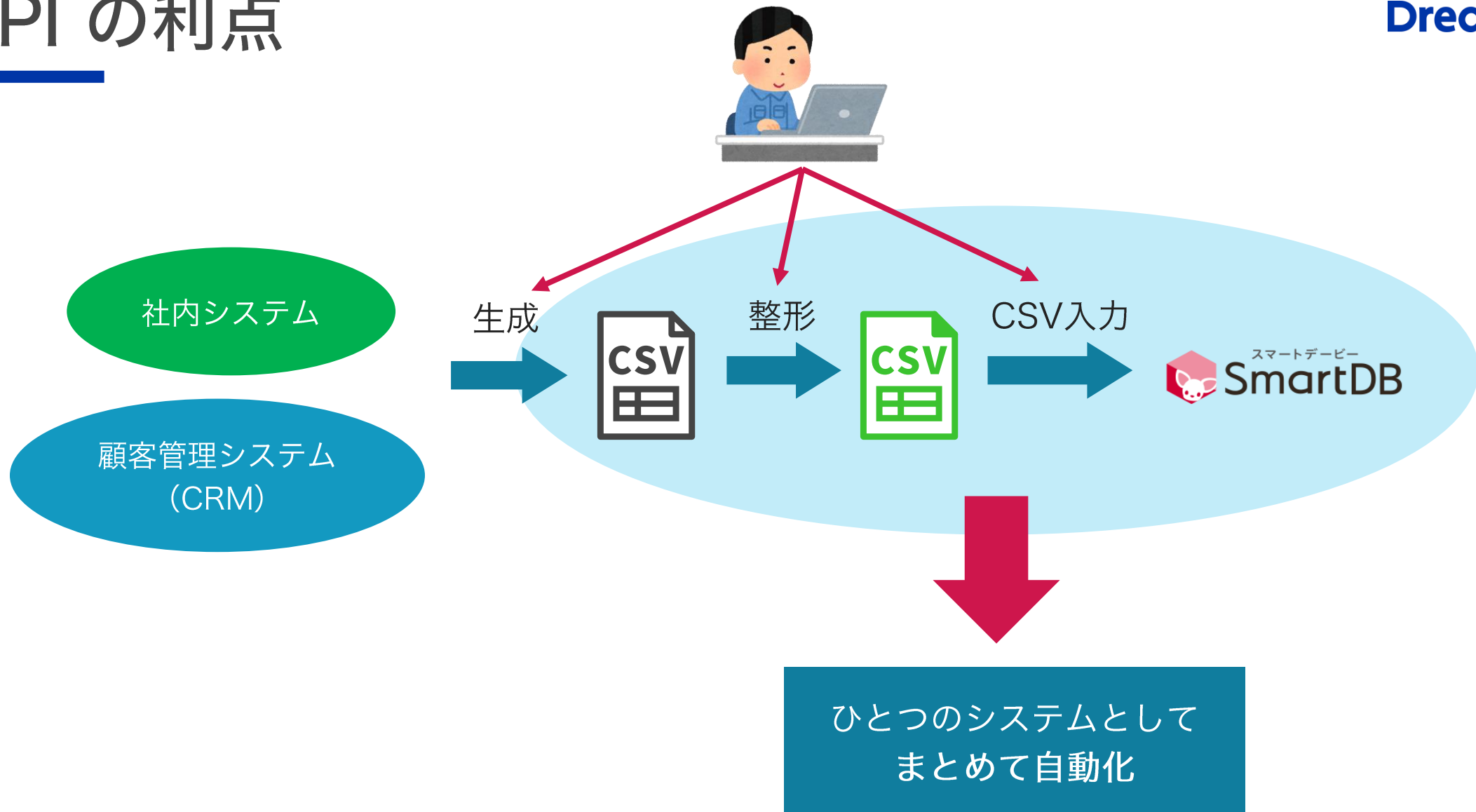


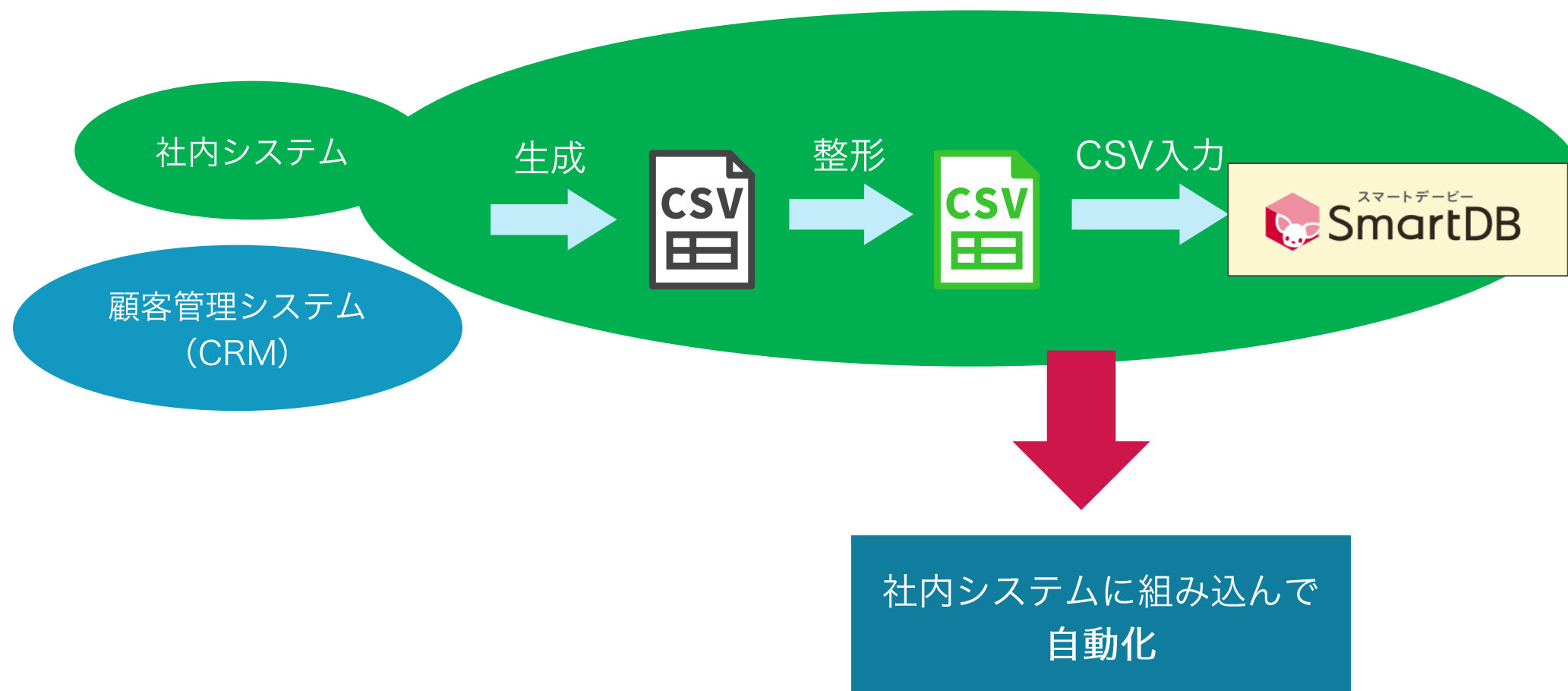
ところで . . .



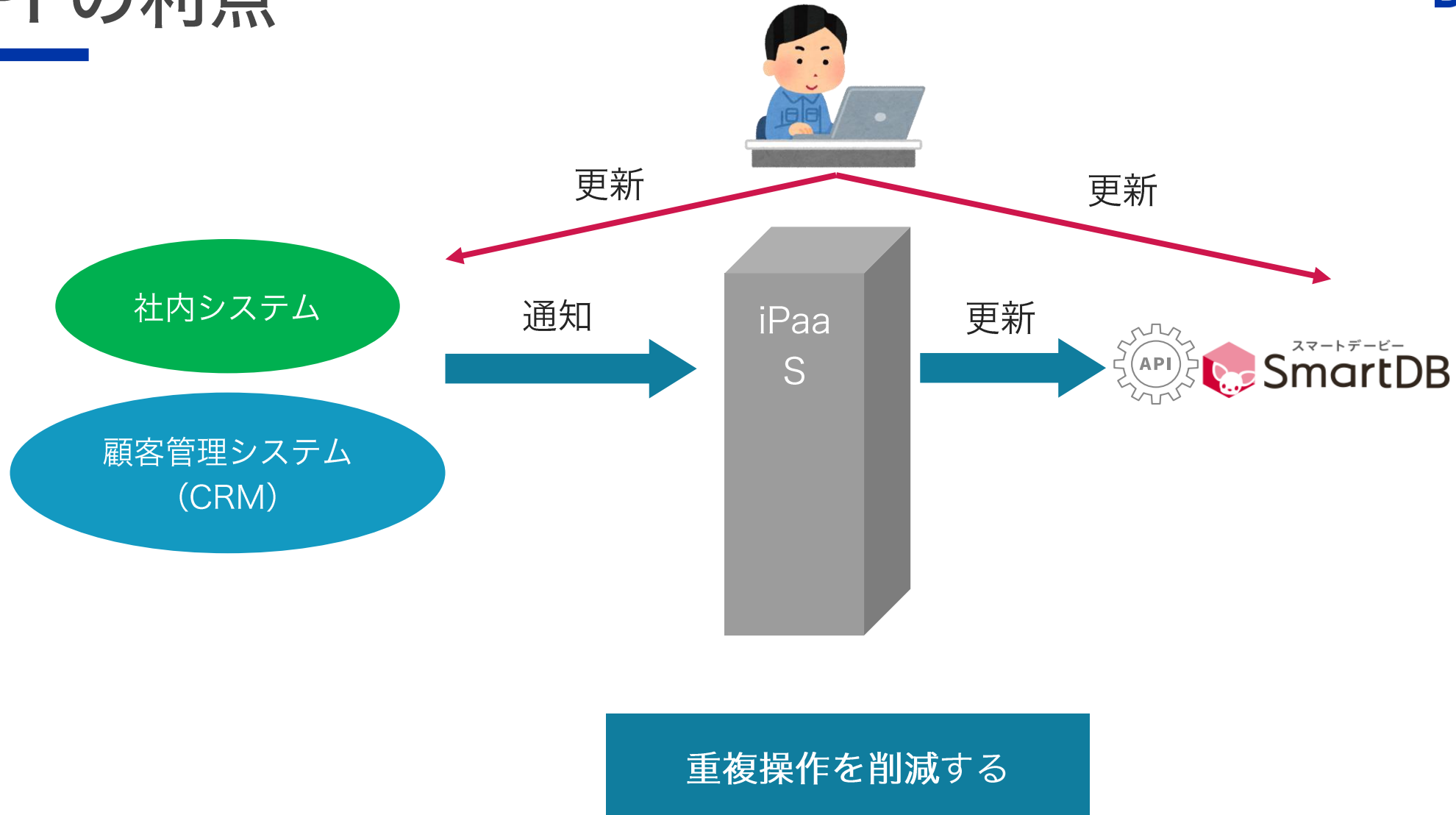
ひとつひとつの操作は SmartDB の画面上からも行うことができる

API の利点





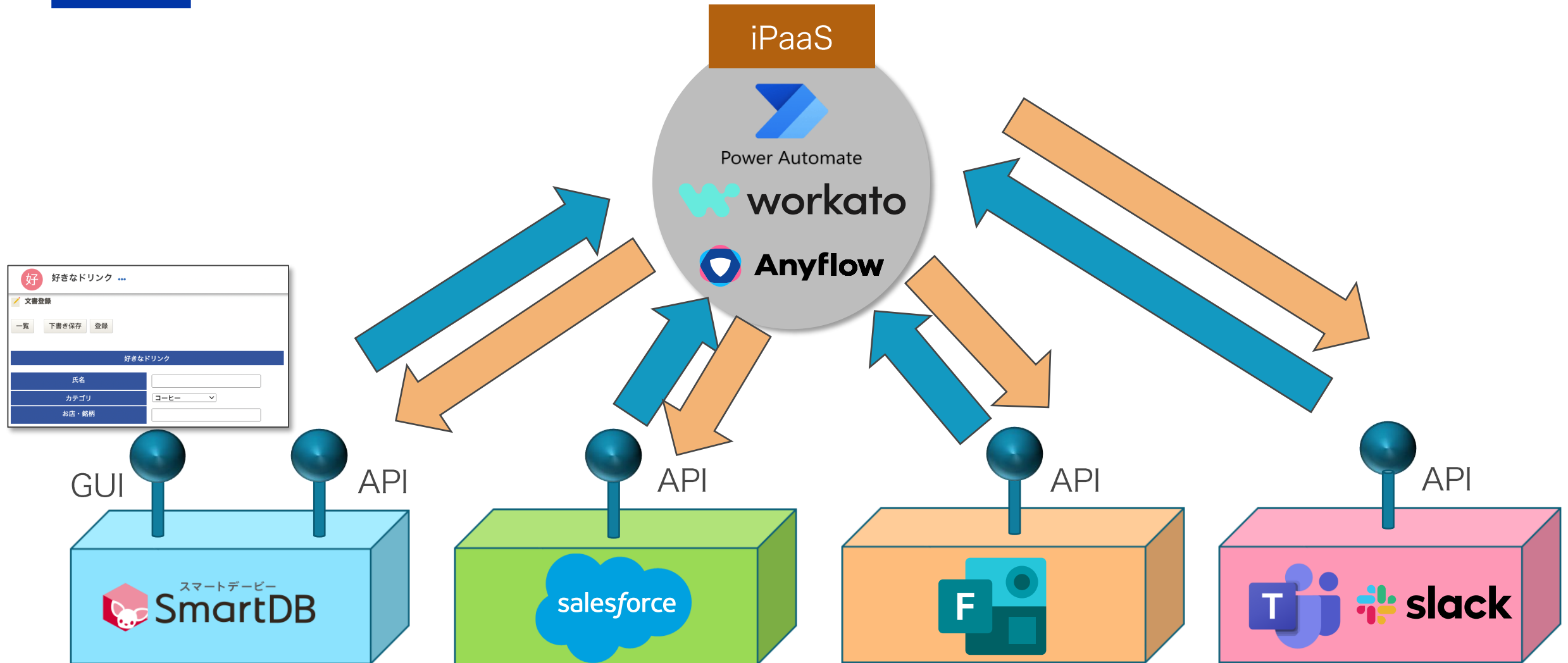
API の利点



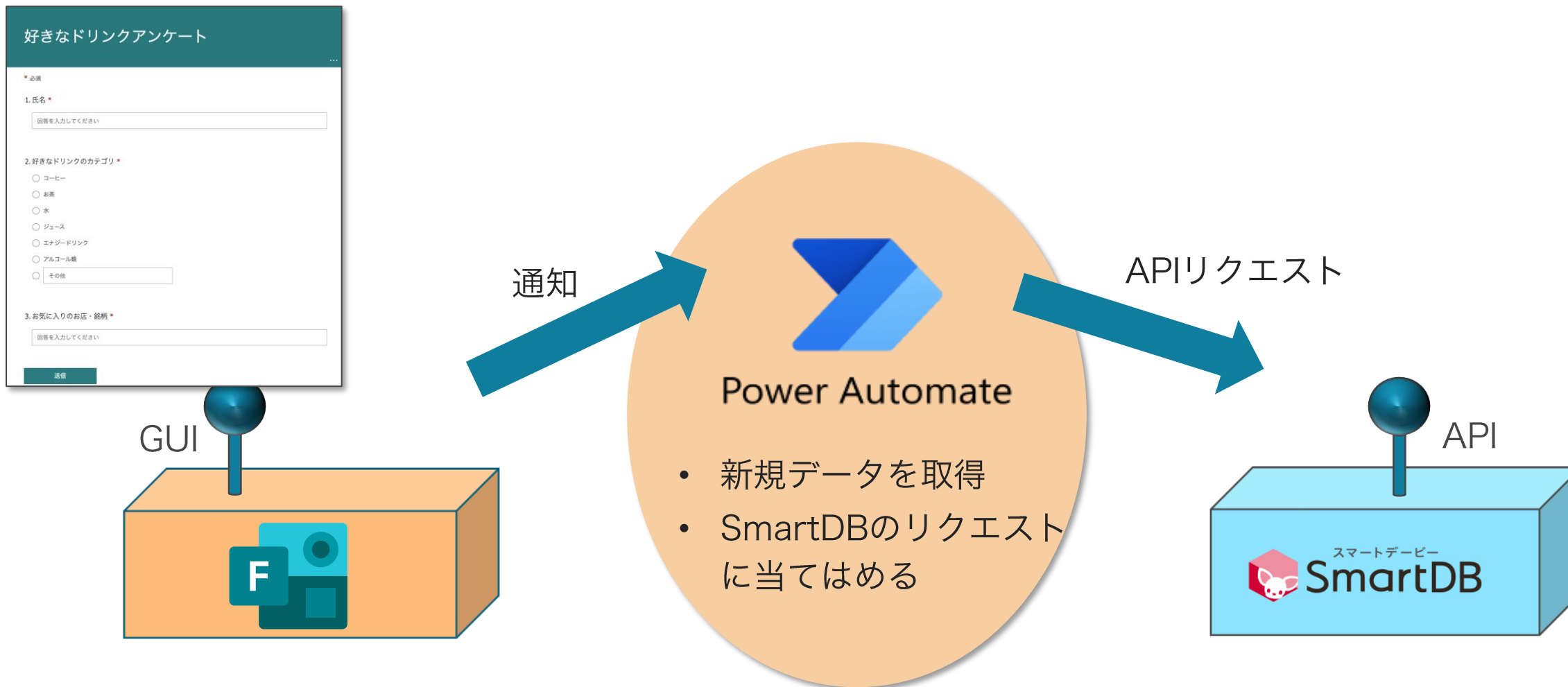
3. 外部連携ツールの 役割について



Web API利用のイメージ



外部サービス連携の例①：アンケート



外部サービス連携の例①：アンケート

新しい応答が送信されるとき

*フォーム ID

↓

応答の詳細を取得する

同じ構造

HTTP

*方法

*URI

ヘッダー

Authorization	Bearer tLW6IXW0mqrIh2MfDLWwtWVixkRcTzC
Content-Type	application/json
キーの入力	値の入力

クエリ

キーの入力	値の入力
-------	------

本文

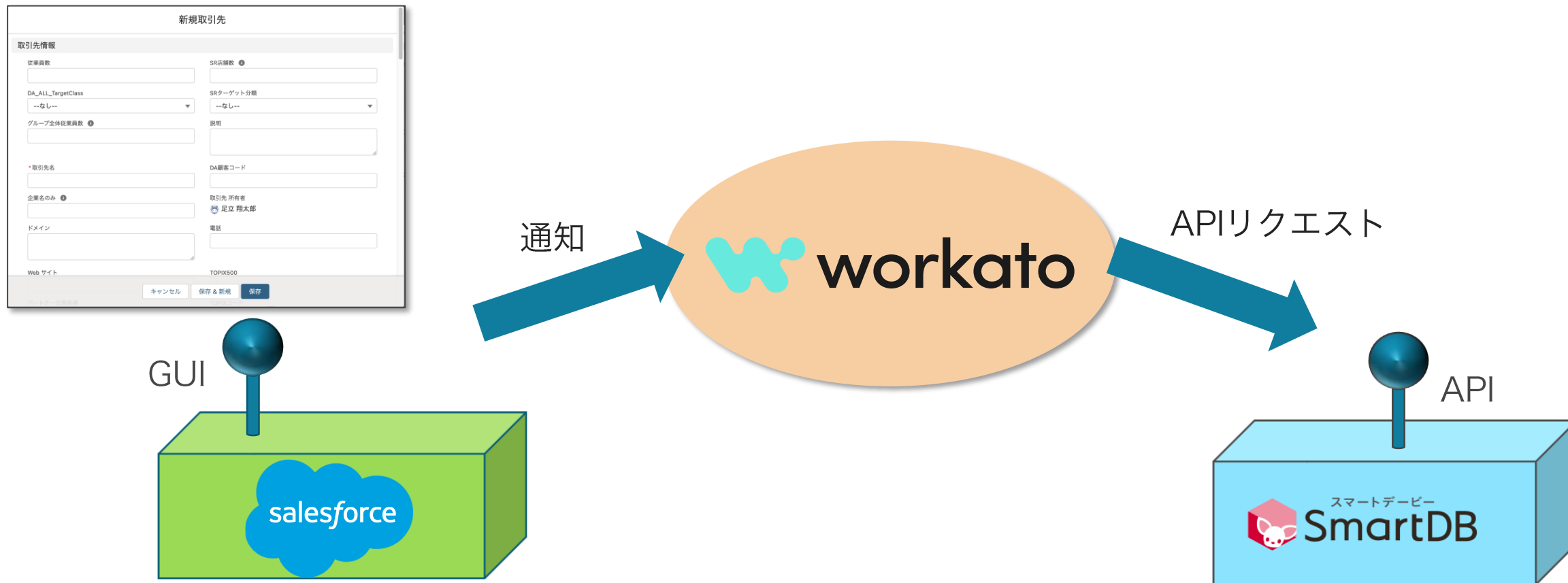
```
{
  "Name": "氏名 ×",
  "Category": "好きなドリンク ×",
  "Brand": "お気に入りの... ×"
}
```

Cookie

詳細オプションを表示する

```
$req = @{
  URI = "https://ae-demo.smartdb.jp/hibiki/rest/2/binders/16219/documents"
  Method = "POST"
  Headers = @{
    Authorization = "Bearer tLW6IXW0mqrIh2MfDLWwtWVixkRcTzC"
    ContentType = "application/json"
  }
  Body = @{
    "Name" = "足立 翔太郎"
    "Category" = 10
    "Brand" = "ゲイシャ"
  }
}
```

外部サービス連携の例②：データの同期

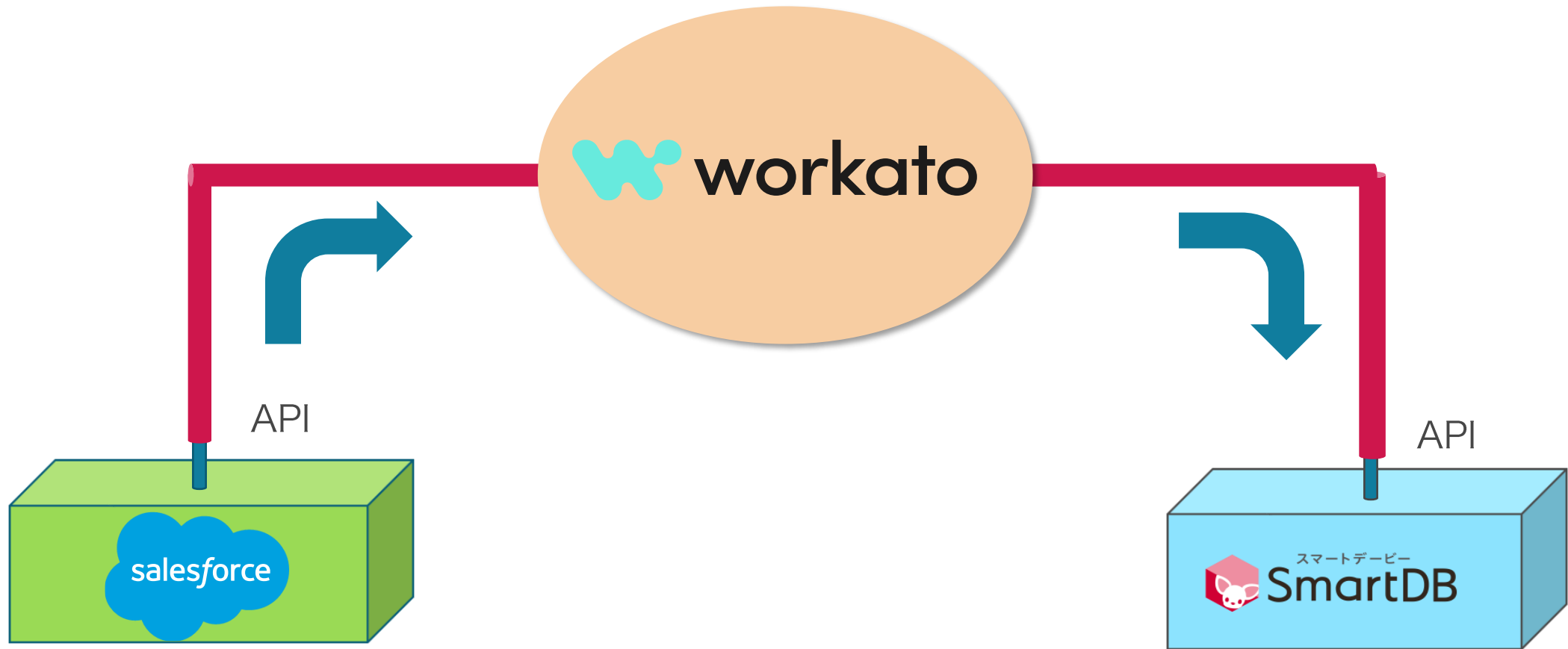


Workato は SmartDB との コネクタ を持っている



SmartDB のバインダの構成などを自動で読み取って
リクエスト内のデータ入力を容易にしてくれるもの

外部サービス連携の例②：データの同期



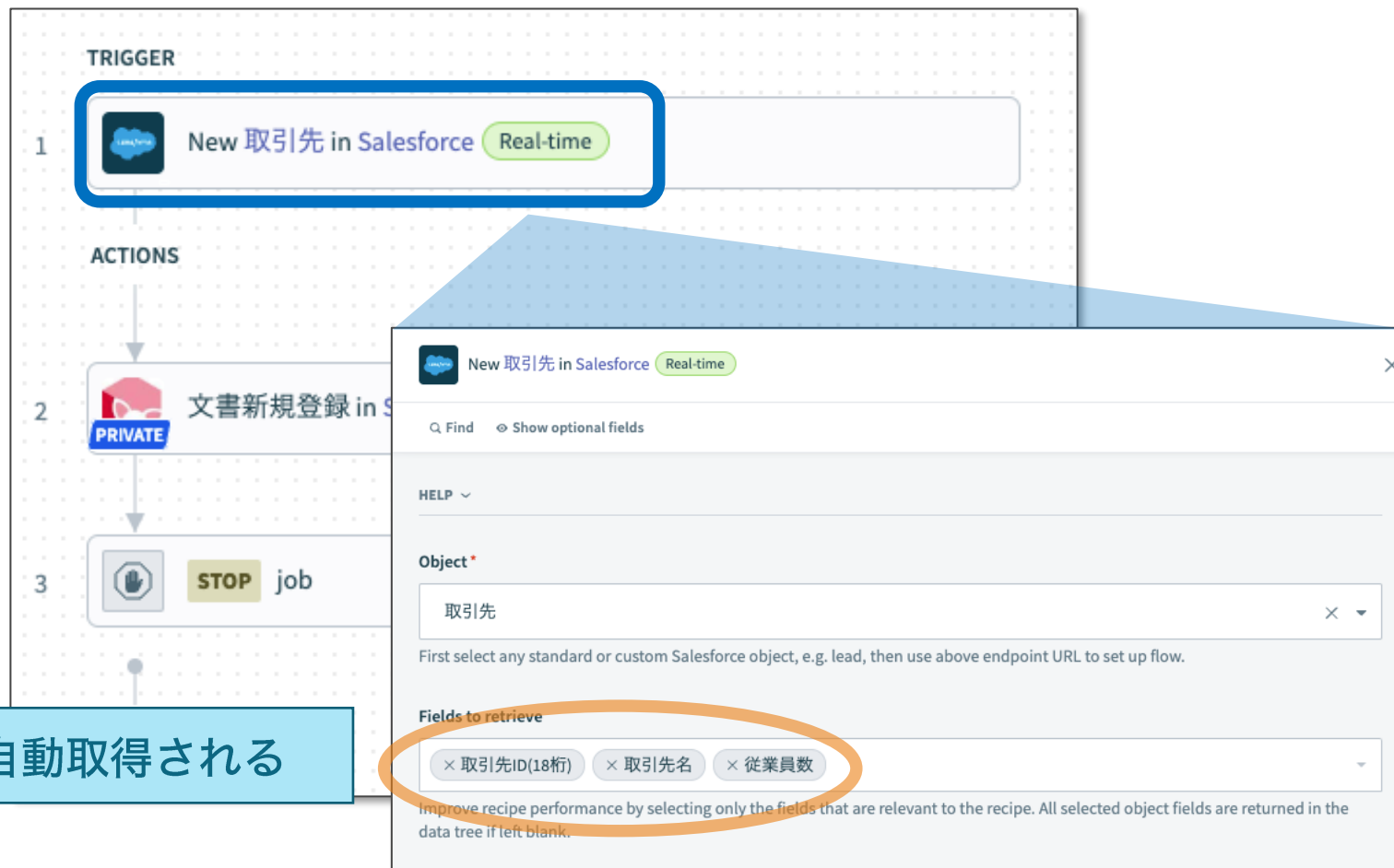
外部サービス連携の例②：データの同期

- Workatoの設定



外部サービス連携の例②：データの同期

● Workatoの設定



コネクタによって自動取得される

外部サービス連携の例②：データの同期

● Workatoの設定

The image shows a Workato workflow and its configuration for a database action. The workflow consists of three steps:

- TRIGGER:** 1. New 取引先 in Salesforce (Real-time)
- ACTIONS:** 2. 文書新規登録 in Smart DB (highlighted with a blue box)
- ACTIONS:** 3. STOP job

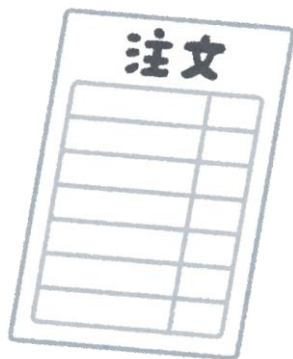
The configuration for the '文書新規登録 in Smart DB' action is shown on the right. It includes the following fields:

- Binder*:** Select from list (dropdown), 取引先マスタ (dropdown)
- Binder (Sample):** Select from list (dropdown), 取引先マスタ (dropdown)
- 取引先ID:** ABC 取引先ID(18桁) | Step 1
- 取引先名:** ABC 取引先名 | Step 1
- 従業員数:** 0.00 従業員数 | Step 1

The fields '取引先ID', '取引先名', and '従業員数' are circled in orange. A blue box at the bottom left of the workflow diagram contains the text: コネクタによって自動取得される

外部サービス連携の例②：データの同期

コネクタのない連携



- メニューをみて自分で書き込む
- 存在しないメニューも書き込める

コネクタのある連携



- 表示されている選択肢から選択
- 存在しないメニューは選べない

APIを使用する際の選択肢比較

種類	操作	Powershell	コネクタのない外部連携	コネクタのある外部連携
共通	操作の決定	○	○	○
	URIの指定	○	○	✓
	認証	○	○	✓
データの取得	必要なデータの選択	○	○	○
	必要なデータの取得	○	○	✓
	SmartDB上の更新を検知	×	○	✓
データの更新	データの選択	○	○	○
	データの成形	○	○	✓

○：自分で設定することで可能

×：難しい

✓：コネクタが自動で行ってくれる

APIを使用する際の選択肢比較

種類	操作	Powershell	Python	REST API	ある外部連携
共通	操作の決定	○	○	○	○
	URIの指定	○	○	○	◎
	認証	○	○	○	◎
データの取得	必要なデータの選択	○	○	○	○
	必要なデータの取得	○	○	○	◎
	SmartDB上の更新を検知	×	○	○	◎
データの更新	データの選択	○	○	○	○
	データの成形	○	○	○	◎

- 実装するためにプログラミングの知識が必須
- 柔軟性が高い

- ：自分で設定することで可能
- ×：難しい
- ◎：コネクタが自動で行ってくれる

APIを使用する際の選択肢比較

種類	操作	Powershell	コネクタのない外部連携	コネクタのある外部連携
共通	操作の決定	○	○	○
	URIの指定	○	○	
	認証	○	○	
データの取得	必要なデータの選択	○	○	
	必要なデータの取得	○	○	◎
	SmartDB上の更新を検知	×	○	◎
データの更新	データの選択	○	○	○
	データの成形	○	○	◎

• API リクエストを作成するために HTTP に関する知識が必須

○：自分で設定することで可能
×：難しい
◎：コネクタが自動で行ってくれる

APIを使用する際の選択肢比較

種類	操作	Powershell	コネクタのない外部連携	コネクタのある外部連携
共通	操作の決定	● ノーコード・ローコード での連携が可能		○
	URIの指定			◎
	認証		○	○
データの取得	必要なデータの選択	○	○	○
	必要なデータの取得	○	○	◎
	SmartDB上の更新を検知	×	○	◎
データの更新	データの選択	○	○	○
	データの成形	○	○	◎

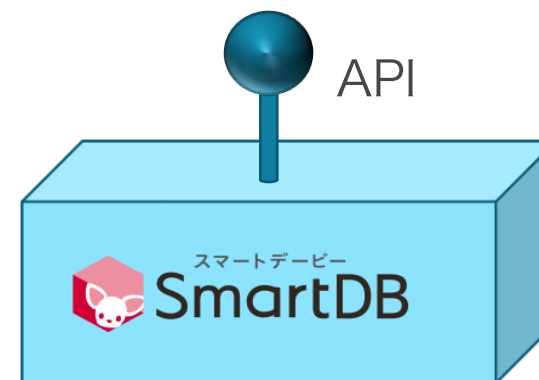
- ：自分で設定することで可能
- ×：難しい
- ◎：コネクタが自動で行ってくれる

4. まとめ



API を業務に活かす

不定期に異なる操作を行う（コマンドライン・プログラム）

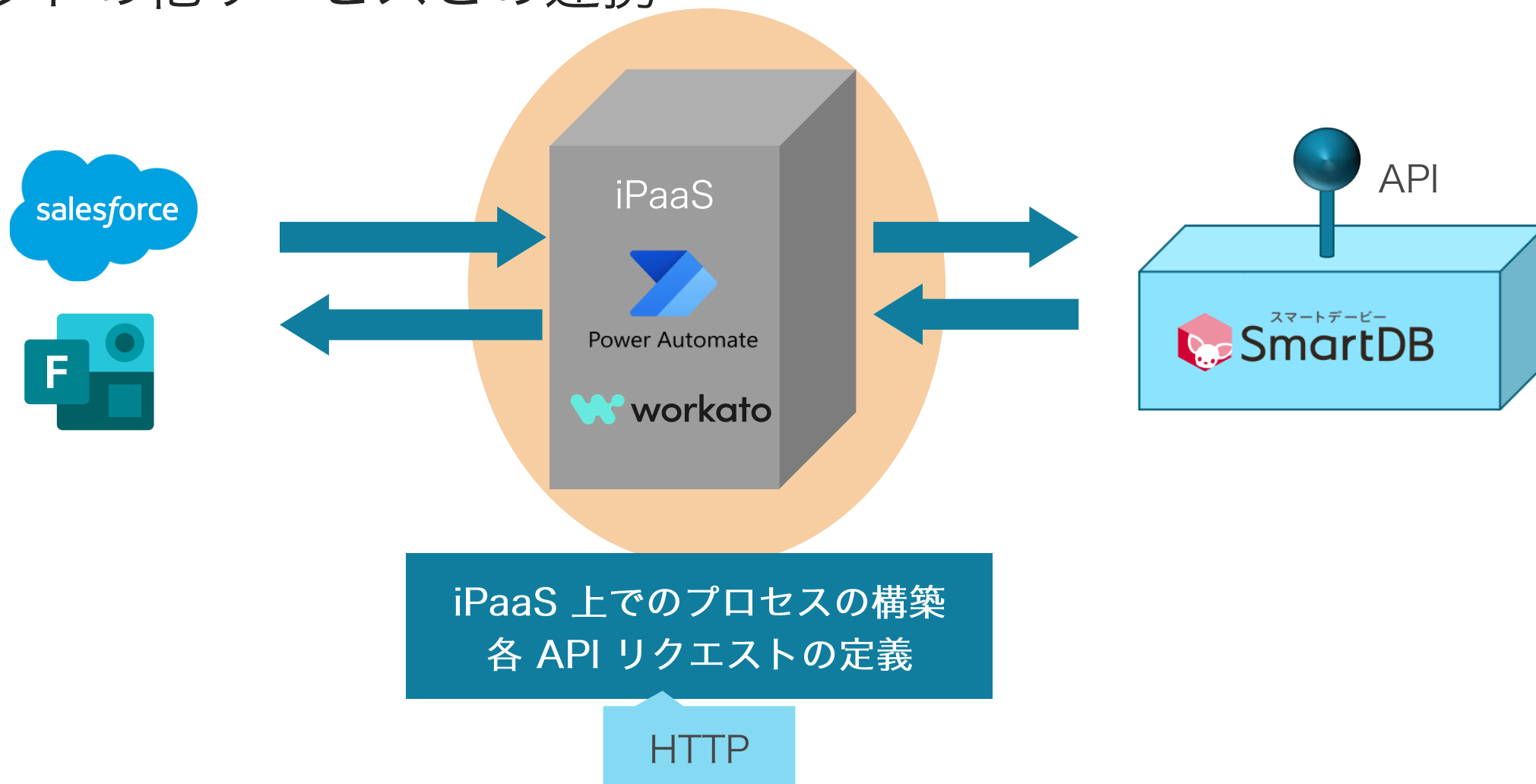


プログラム・コマンドの作成

PowerShell・Bash
JavaScript・Java・
Python etc ..

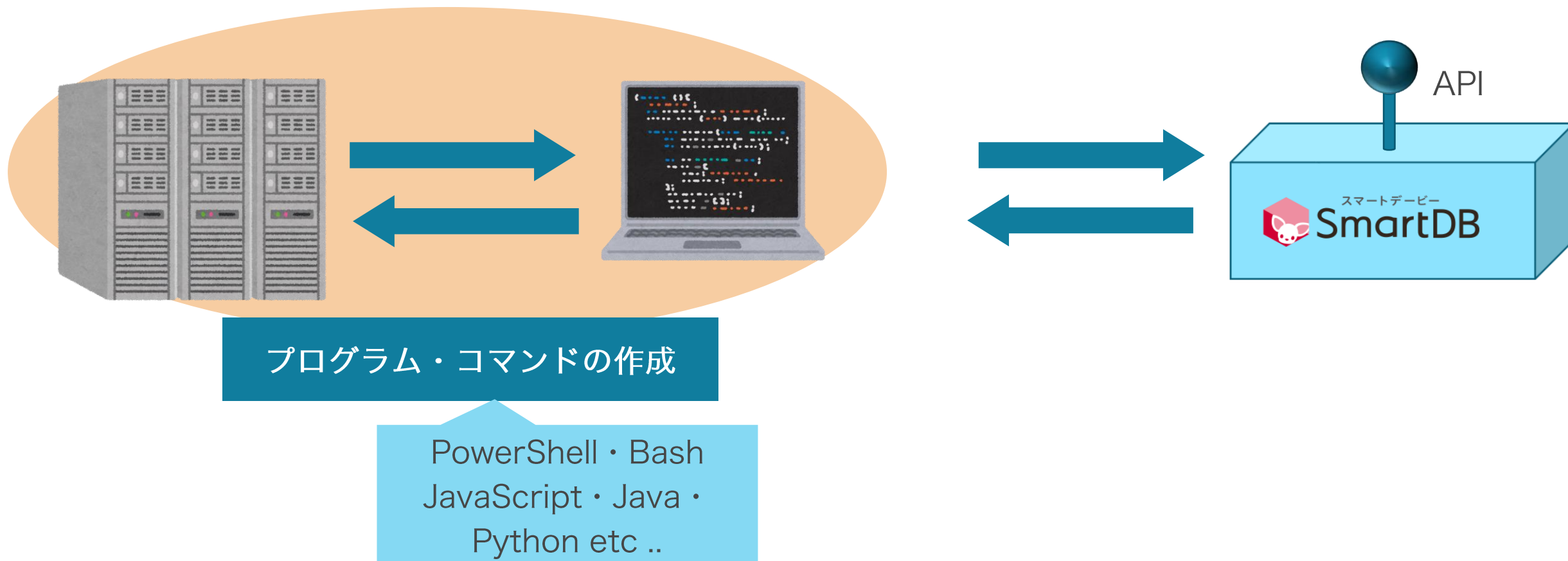
API を業務に活かす

クラウドの他サービスとの連携



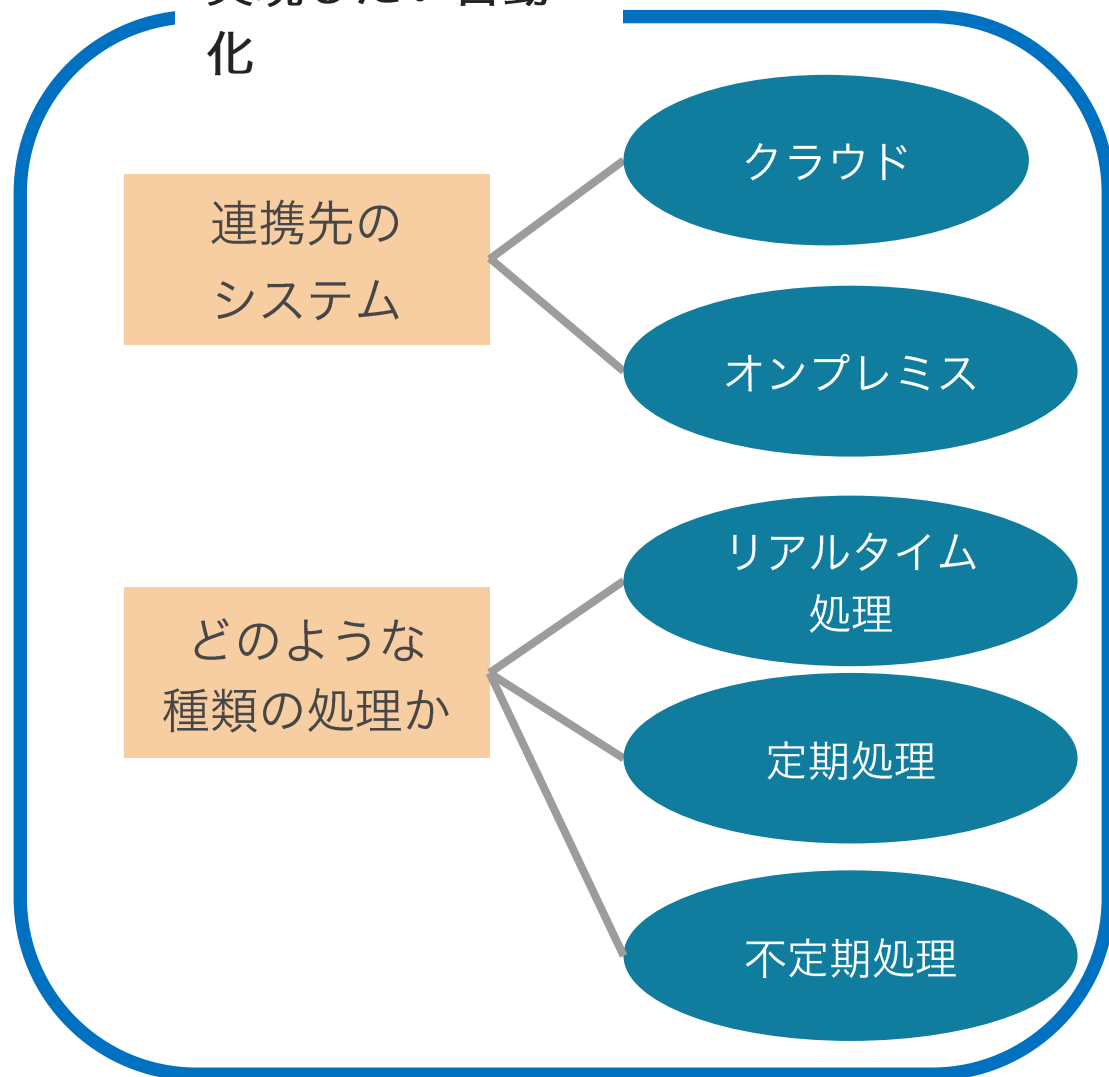
API を業務に活かす

オンプレミスのお他サービス（社内システム）との連携

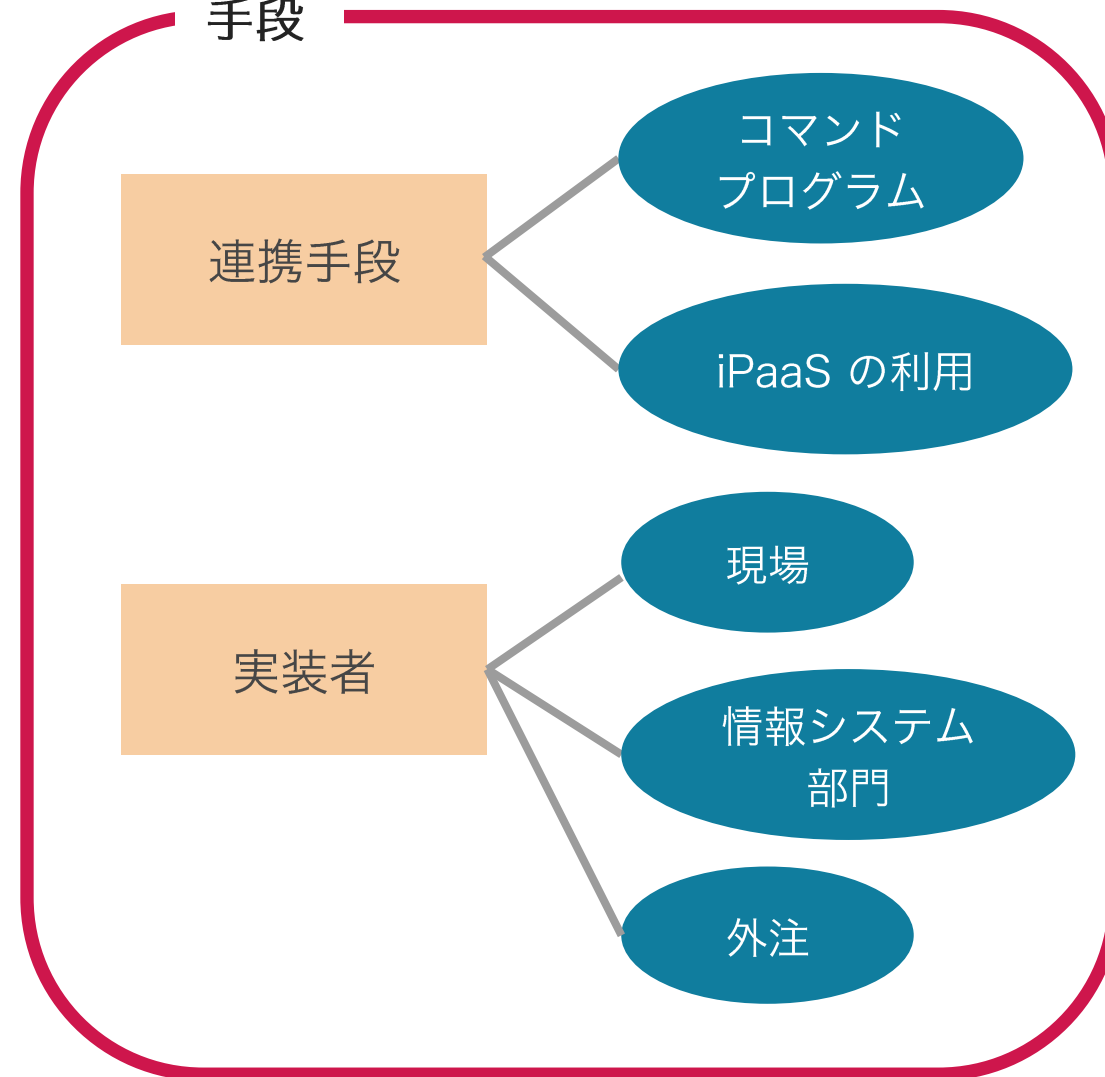


API 連携の導入にあたって

実現したい自動
化



手段



アンケートのお願い

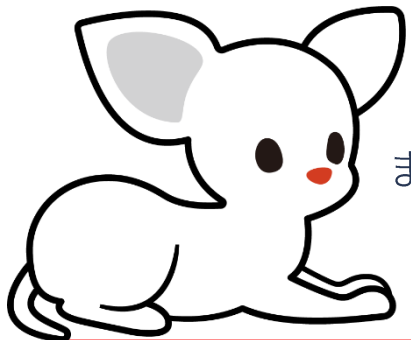


アンケートのお願い

本日はありがとうございました。

今後の改善のためにいろいろご意見頂戴できれば幸いです。

<https://forms.office.com/r/hyM6RqwaF1>



また次回もお楽しみに