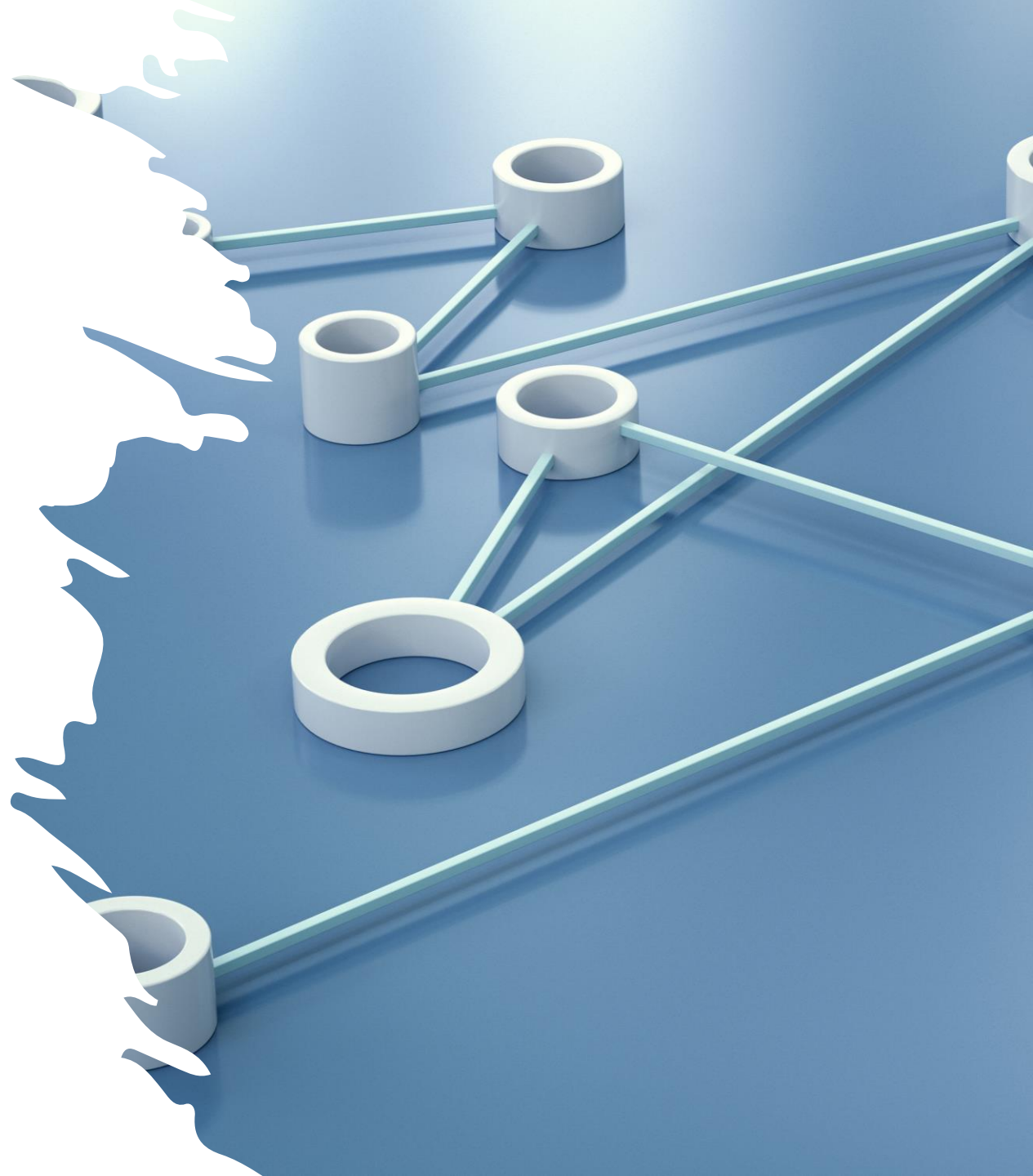


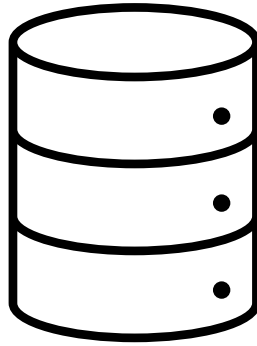
## システムの継承と 新技術の連携によるDX

株式会社ランサ・ジャパン  
技術部 笠原一哲

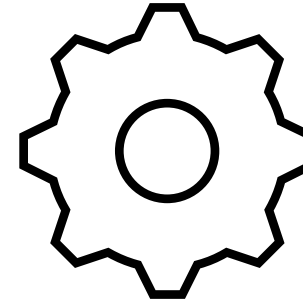


# 既存システムは大きな資産

## 既存システム



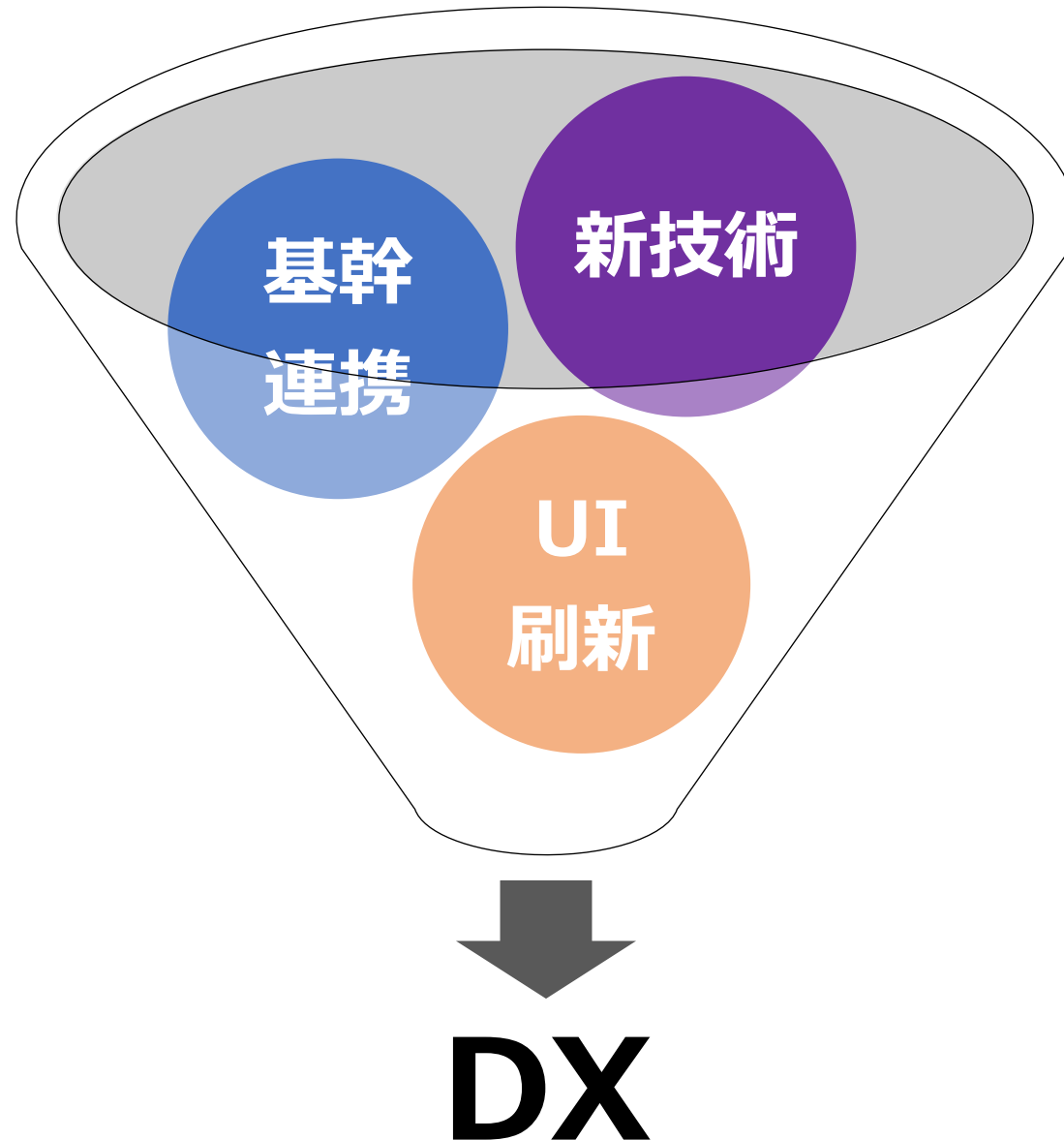
長年のデータ  
DB



業務最適化  
PGM

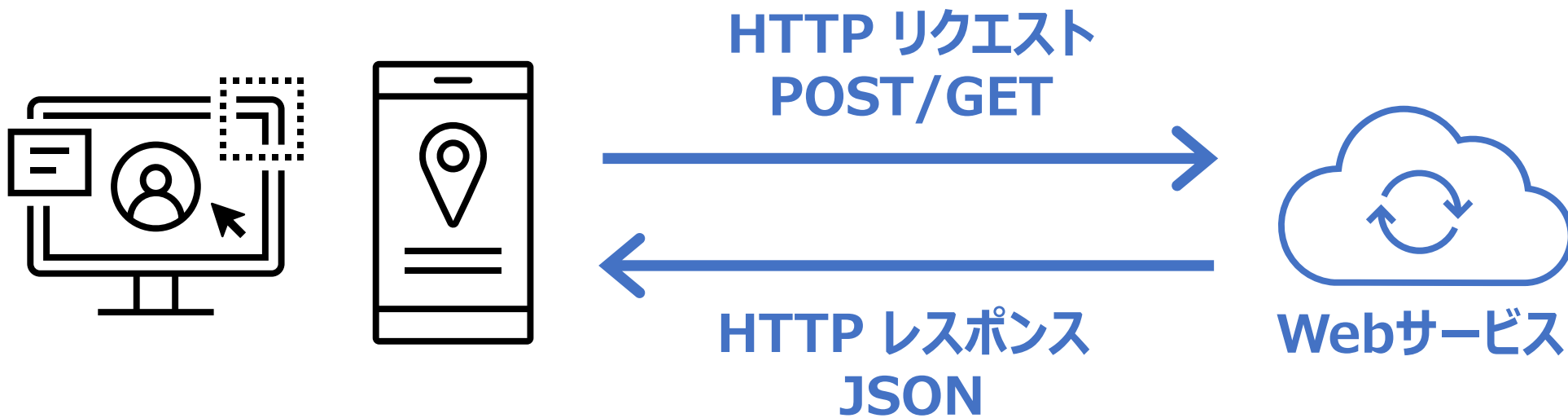


# 基幹連携とDX推進

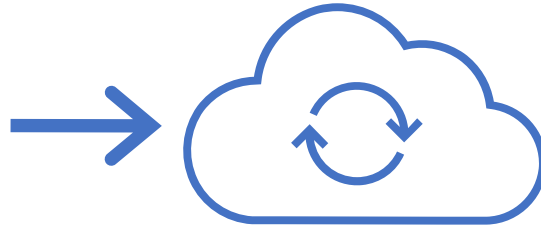
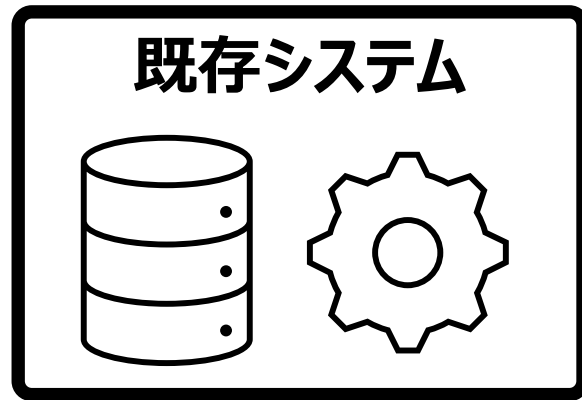


# Webサービス (REST API)

共通仕様  
OSやプログラムを問わない



# Webサービスの実行



# Webサービスの活用例

## シングル・サインオン


Login to continue


Username


Password

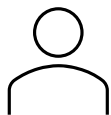
**LOGIN**

**REGISTER**

 Login with Google

 Login with Microsoft

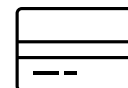
 Login with Facebook



## リアルタイムの情報取得



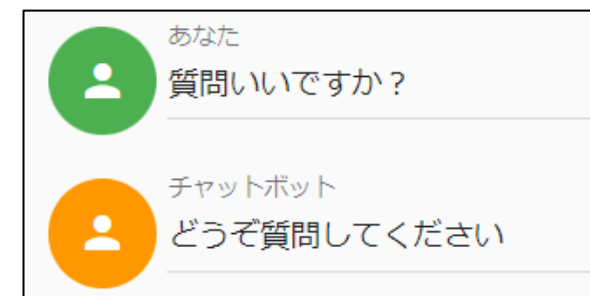
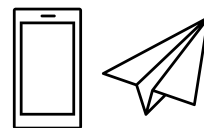
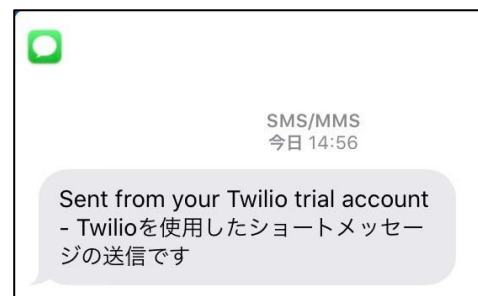
	USD	EUR	GBP	JPY
USD	1	0.901306	0.77808	108.949977
EUR	1.1095	1	0.86328	120.88
GBP	1.285214	1.158372	1	140.024094



Tokyo (Narita International)		
Arrivals		
Airline	Destination	Time
 DELTA	Honolulu	15:06
 JAPAN AIRLINES	Manila	15:08
 CHINA EASTERN	Shanghai	15:12



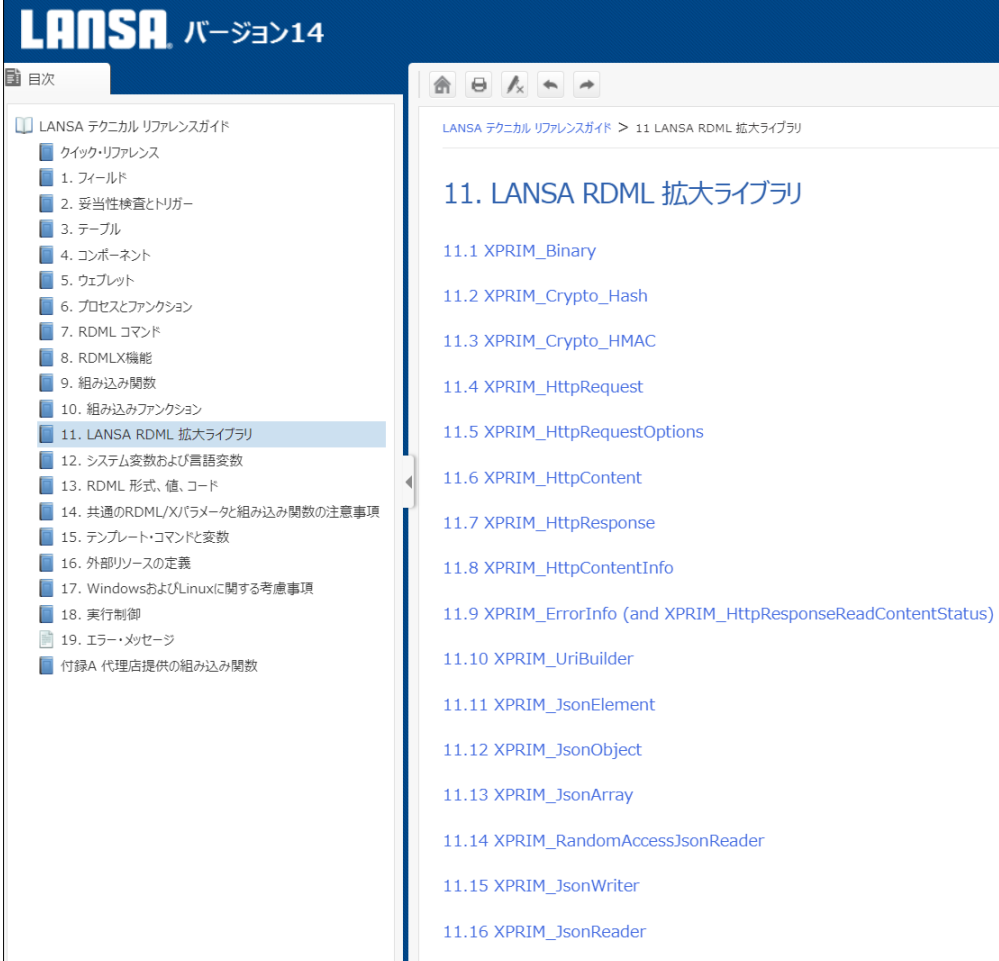
## コミュニケーション・ツールの連携



# LANSAでWebサービス実行 コンポーネント

機能	コンポーネント
URL エンコード	XPRIM_UriBuilder
HTTP メソッド	XPRIM_HttpRequest
JSON パーサー	XPRIM_RandomAccessJsonReader
バイナリー処理	XPRIM_BINARY
ハッシュ計算	XPRIM_Crypto_Hash
...	

## LANSA テクニカル リファレンスガイド



LANSA バージョン14

目次

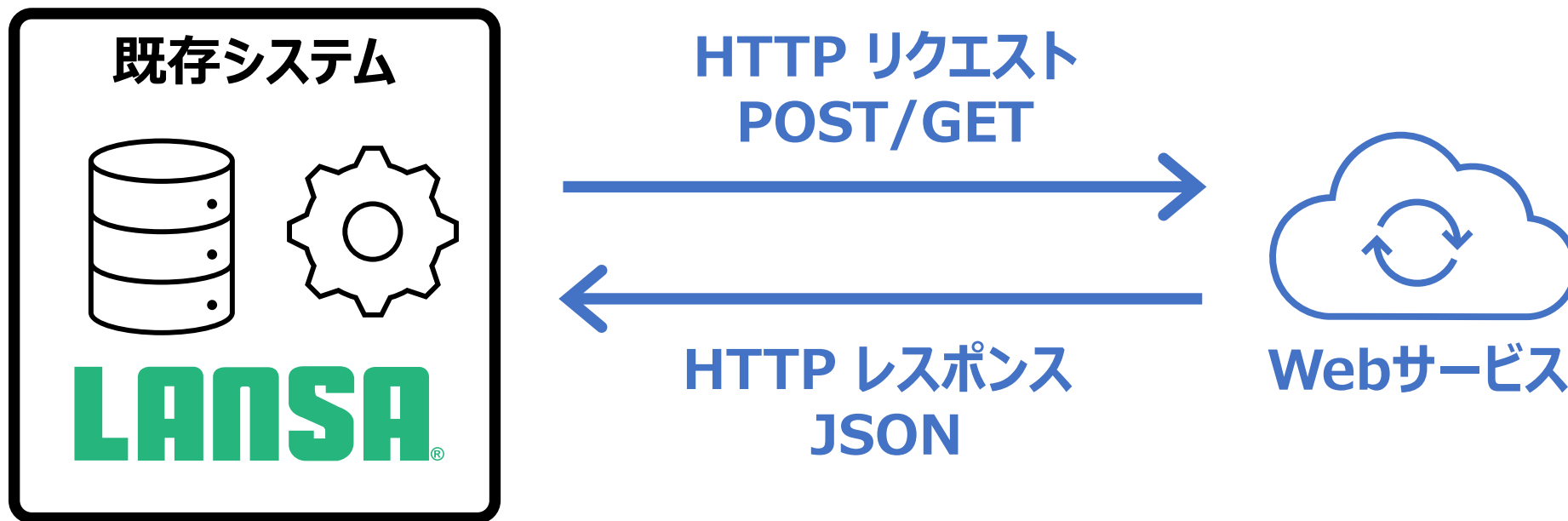
- LANSA テクニカル リファレンスガイド
  - クイック・リファレンス
  - 1. フィールド
  - 2. 妥当性検査とトリガー
  - 3. テーブル
  - 4. コンポーネント
  - 5. ウェブレット
  - 6. プロセスとファンクション
  - 7. RDML コマンド
  - 8. RDMLX機能
  - 9. 組み込み関数
  - 10. 組み込みファンクション
  - 11. LANSA RDML 拡張ライブラリ**
  - 12. システム変数および言語変数
  - 13. RDML 形式、値、コード
  - 14. 共通のRDML/Xパラメータと組み込み関数の注意事項
  - 15. テンプレート・コマンドと変数
  - 16. 外部リソースの定義
  - 17. WindowsおよびLinuxに関する考慮事項
  - 18. 実行制御
  - 19. エラー・メッセージ
  - 付録A 代理店提供の組み込み関数

LANSA テクニカル リファレンスガイド > 11 LANSA RDML 拡張ライブラリ

### 11. LANSA RDML 拡張ライブラリ

- 11.1 XPRIM\_Binary
- 11.2 XPRIM\_Crypto\_Hash
- 11.3 XPRIM\_Crypto\_HMAC
- 11.4 XPRIM\_HttpRequest
- 11.5 XPRIM\_HttpRequestOptions
- 11.6 XPRIM\_HttpContent
- 11.7 XPRIM\_HttpResponse
- 11.8 XPRIM\_HttpContentInfo
- 11.9 XPRIM\_ErrorInfo (and XPRIM\_HttpResponseReadContentStatus)
- 11.10 XPRIM\_UriBuilder
- 11.11 XPRIM\_JsonElement
- 11.12 XPRIM\_JsonObject
- 11.13 XPRIM\_JsonArray
- 11.14 XPRIM\_RandomAccessJsonReader
- 11.15 XPRIM\_JsonWriter
- 11.16 XPRIM\_JsonReader

# LANSAでWebサービス実行 概要





# LANSAでWebサービス実行 POST実行



## \* WebサービスのURLを作成

```
Define_Com Class(#XPRIM_UriBuilder) Name(#UrlBuilder)  
#UrlBuilder.SetScheme('https')  
#UrlBuilder.SetHost('ホスト名')  
#UrlBuilder.SetPath('パス')
```

## \* キーと値を設定

```
Define_Com Class(#XPRIM_HttpRequest) Name(#Request)  
#Request.Content.AddUrlEncodedFormValue Name('キー') Value('値')
```

## \* 実行

```
#Request.DoPost Url(#UrlBuilder)
```

# LANSAでWebサービス実行 JSON取得



\* JSONを読むコンポーネントを定義

```
Define_Com Class(#XPRIM_RandomAccessJsonReader) Name(#Reader)
```

\* 絶対パスで取得

```
#VALUE1 := #Reader.ReadStringWithPath("twilio_services/accountSid")
```

\* 相対パスで取得

```
#Reader.BeginObjectWithPath("twilio_services")
```

```
#VALUE1 := #Reader.ReadStringWithPath("accountSid")
```

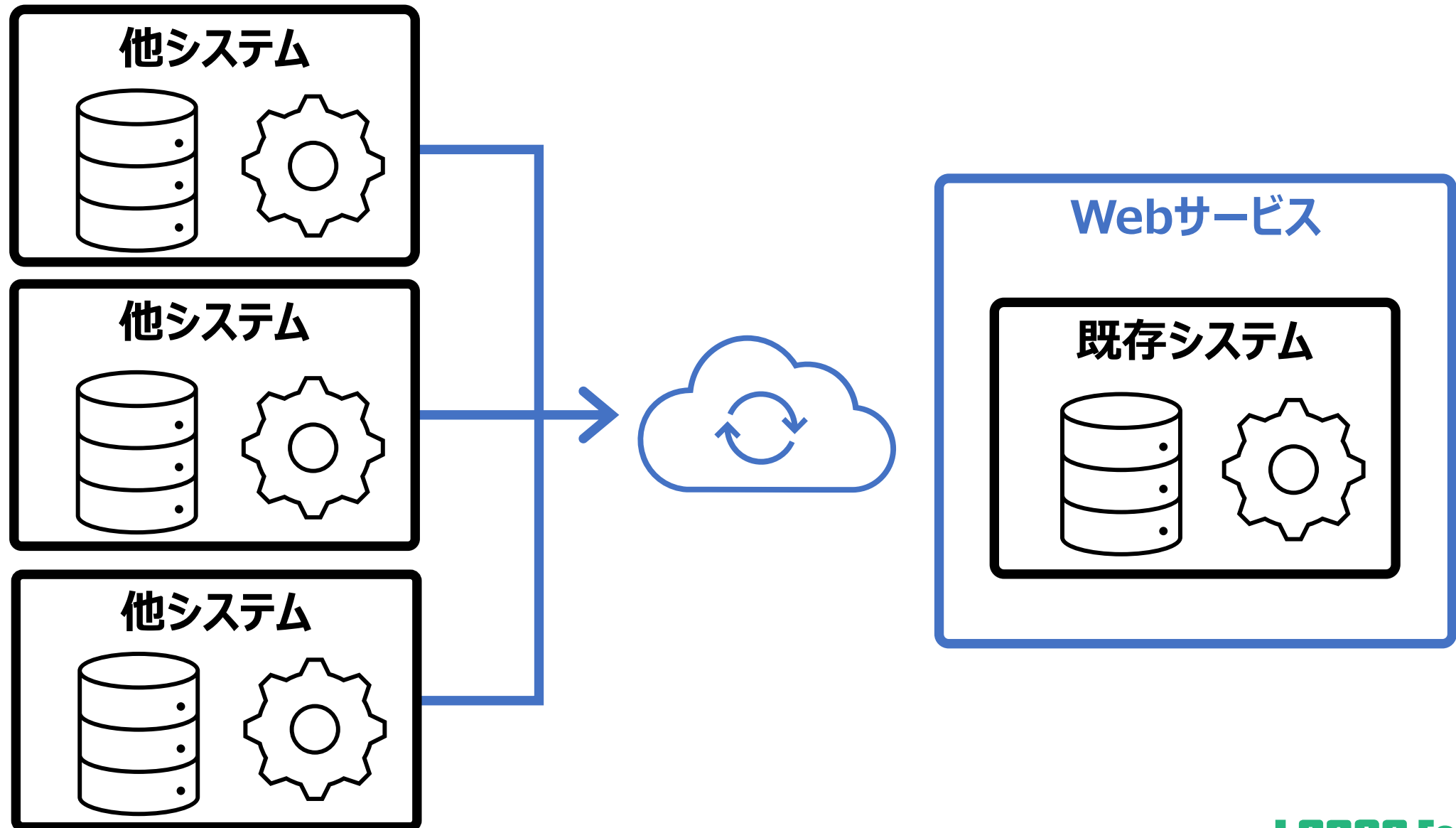
# LANSAでWebサービス実行 サンプル

## Webサービス

機能	Webサービス
シングル サインオン	OAuth2
スパム/ボット対策	Google reCAPTCHA
地図/位置情報	Google Geocoding
翻訳	Google Translate
クラウド ストレージ	AWS S3
ショート メッセージ	Twilio
決済	PayPal
メール	SendGrid
気象	OpenWeatherMap
...	

LANSA バージョン14	
目次	
Webサービス	
JSON データの作業	
Webサービスの利用	
ソフトウェア 前提条件	
IBM i の制限事項	
RESTful Web とは?	
Web サービスの起動はクライアント側? サーバー側?	
XPRIM_HttpRequest の利用	
HTTP 要求の設定	
XPRIM UriBuilder を使った URL の構築	
要求ヘッダーの追加	
要求本文へのコンテンツ追加	
要求本文用の JSON データの構築	
複数パート (Multipart) 本文の作成	
応答の読み込み	
XPRIM_HttpRequest オブジェクトの再利用	
要求・応答データのログ	
チュートリアル 1 - Google 翻訳 API	
チュートリアル 2 - Web サービス経由で Java コードを実行	
チュートリアル 3 - Web サービスを利用した .NET との相互運用性	
Web API クライアントのサンプル・ライブラリ	

# 既存システムのWebサービス化



# LANSAでWebサービス作成 サーバーモジュール



新規サーバー モジュール

名前: LAPI 作成(C)

記述: LANSA Restful API キャンセル(N)

識別子: LAPI

プライマリ テーブル

オブジェクト名称: PSLMST

テーブル名: PSLMST \*\*\*

ライブラリ: V50KASDTA

セカンダリ テーブル (任意)

オブジェクト名称:

テーブル名: \*\*\*

ライブラリ: V50KASDTA

セキュリティ (オプション)

セキュリティ タイプ: JWT as bearer Token

☒ セキュリティ サンプル生成

☒ CRUD に適用

リポジトリ

リポジトリ

アイテム 記述

- I
- J
- K
- L
- M
- N
- O
- P
  - PSLDEV 社員支給スマートデバイス I D
  - PSLDSE DataStore Events
  - PSLEVENT 社員イベント ログ
  - PSLIMG 社員画像
  - PSLMST 社員マスター
  - PSLSKL 社員資格情報ファイル
  - PSLTIME 出退勤ファイル
  - PSLTIMES 人事タイムシート
- Q
- R
- S
- T
- U

ソース API 定義 リポジトリ詳細 クロスリファレンス

定義

サービス名: /LAPI

区画処理: パス変数

言語処理:

Swagger V2 として公開: Yes

オープン Api V3 として公開: Yes

情報

セキュリティ要件

パス

コンポーネント

スキーマ

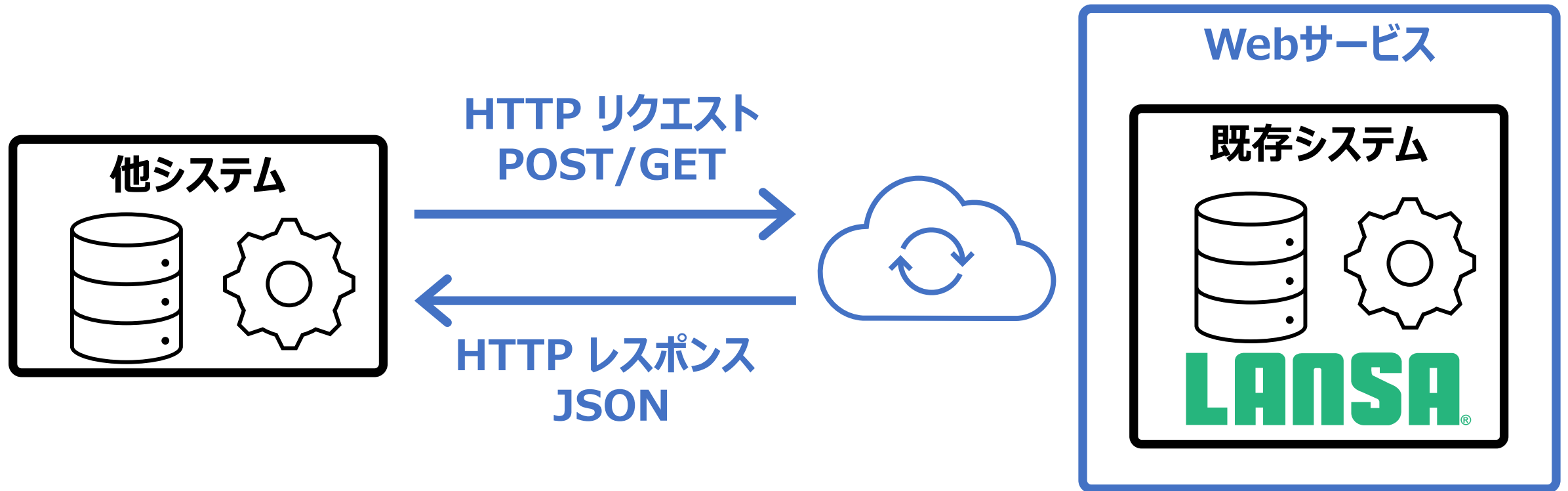
タイプ

- PSLMSTArray
- PSLMSTObject
  - タイトル
  - 記述: PSLMST Data Type
  - タイプ: Object
  - Null 許容: No

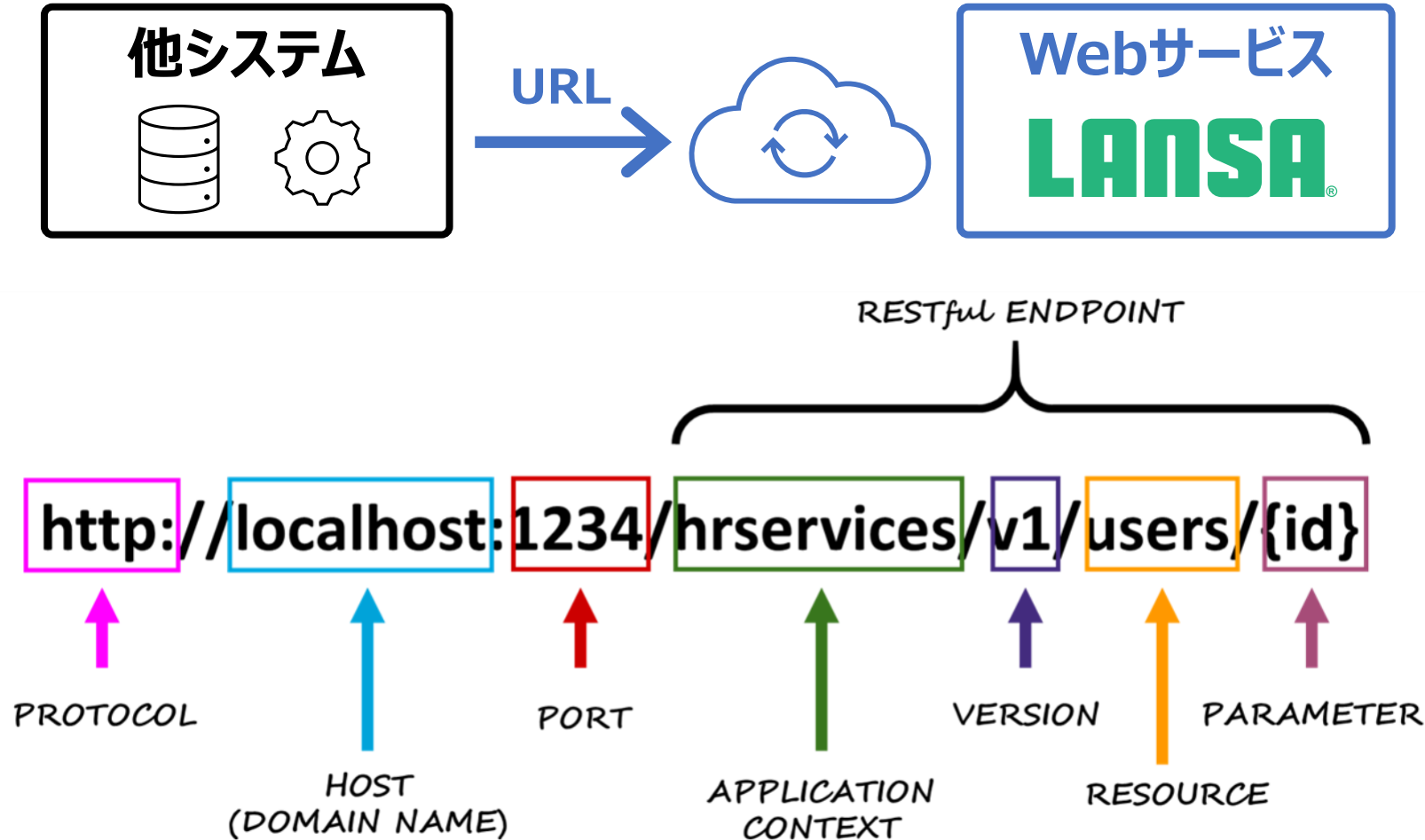
プロパティ

- EMPNO
- SURNAME
- GIVENAME
- ADDRESS1
- ADDRESS2
- ADDRESS3
- POSTCODE
- PHONEHME

# LANSAでWebサービス作成 概要



# LANSAでWebサービス作成 URLとプログラム



# LANSAでWebサービス作成 API定義 1

## ●パス

- ・ 実行URL
- ・ `http://サーバー名:ポート/パス名/{パラメーター名}`
- ・ 例: `LJREST/{EMPNO}`

## ●パラメーター

- ・ 実行URLに指定するパラメーター名
- ・ 例: `EMPNO`

## ●スキーマオブジェクト

- ・ パラメーターの属性 (タイプ、長さ)
- ・ 例: `Type String`

ソース		API 定義		リポジトリ詳細		クロスリファレンス	
パス							
▼ /LJREST/{EMPNO}							
概要							
詳細							
▼ パラメータ							
▼ EMPNO							
識別子							
記述						EMPNO	
位置						Path	
スタイル						Default	
必須						Yes	
展開						No	
廃止予定						No	
予約を許可						No	
空の値を許可						No	
▼ スキーマ オブジェクト							
タイトル						EMPNO	
記述						EMPNO	
タイプ						String	
形式						Unspecified	



# LANSAでWebサービス作成 API定義 2

## ●オペレーション

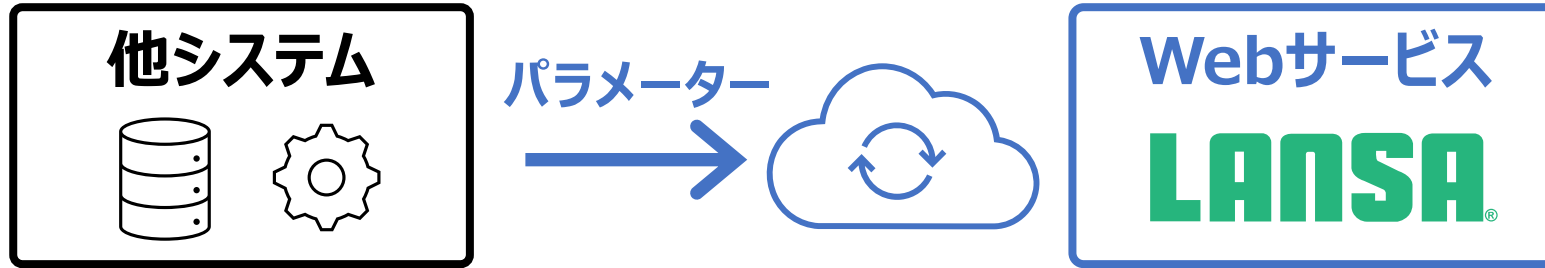
- ・ 動詞
  - ・ 実行方法 (HTTPメソッド等)
- ・ サーバルーチン
  - ・ 実行するサーバルーチン名

## ●応答

- ・ 状態コード
  - ・ 例: 200
- ・ メディア タイプ
  - ・ 例: application/json

パス	
/LIREST/{EMPNO}	
概要	
詳細	
パラメータ	
オペレーション	
GetEmployee	
動詞	Get
サーバ ルーチン	GetEmployee
概要	
詳細	
トップレベルのセキュリティを無視	No
廃止予定	No
パラメータ	
要求	
応答	
OK	
状態コード	200
メディア タイプ	
application/json	

# LANSAでWebサービス作成 パラメーター取得



\* Webサービス用の設定値を指定

Srvroutine Name(GetEmployee) Response(\*HTTP #context)

\* パラメーターEMPNOの値を#EMPNOにセット

#context.Request.PathParameters.TryGetValue( "EMPNO", #EMPNO )

Fetch Fields(\*ALL) From\_File(PSLMST) With\_Key(#EMPNO)

...

# LANSAでWebサービス作成 JSON作成



## \* JSON作成のコンポーネントを定義

```
Define_Com Class(#PRIM_IOC.StringWriter) Name(#StringWriter)
```

```
Define_Com Class(#PRIM_JSON.Writer) Name(#Writer) TextWriter(#StringWriter)
```

## \* JSON作成 (キー LastName 値 #GIVENAME キー FirstName 値 #SURNAME)

```
#Writer.BeginObject
```

```
#Writer.WriteString( #GIVENAME, "LastName" )
```

```
#Writer.WriteString( #SURNAME, "FirstName" )
```

```
#Writer.EndObject
```

# LANSAでWebサービス作成 レスポンス



\* HTTP ステータス

```
#context.Response.HttpStatus := 200
```

\* メディア タイプ

```
#context.Response.ContentType := "application/json"
```

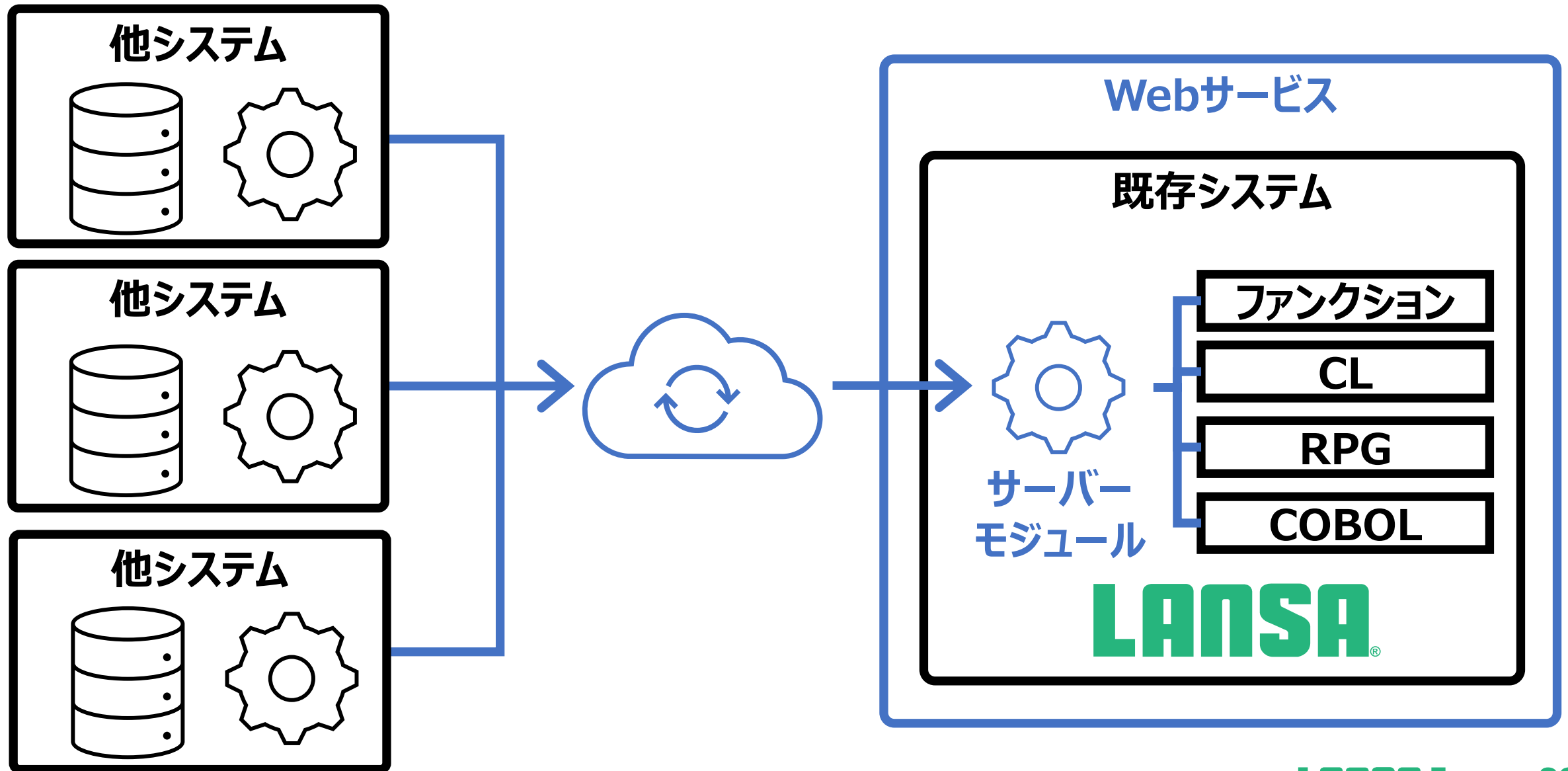
\* HTTP レスponse ボディ

```
#context.Response.ContentString := #StringWriter.Text
```

Return

Endroutine

# LANSAでWebサービス作成 既存PGMの活用



# LANSAでWebサービス作成 マニュアル



## Publish Web Services

☰ Publish Web Services

🏠 Home

REST API ^

★ API Basics

★ URL Anatomy

★ API Definition in VL

Initial Setup ^

☰ Prerequisites

⚙️ Partition Setup

TUTORIALS ^

📖 Server Module

Single Resource v

Multiple Resources v

API Errors v

📄 Downloads

Initial Setup

1 —  
2 —  
3 —

Run Demo

🔌

Downloads

↓

Tutorials

➡

REST API Concepts

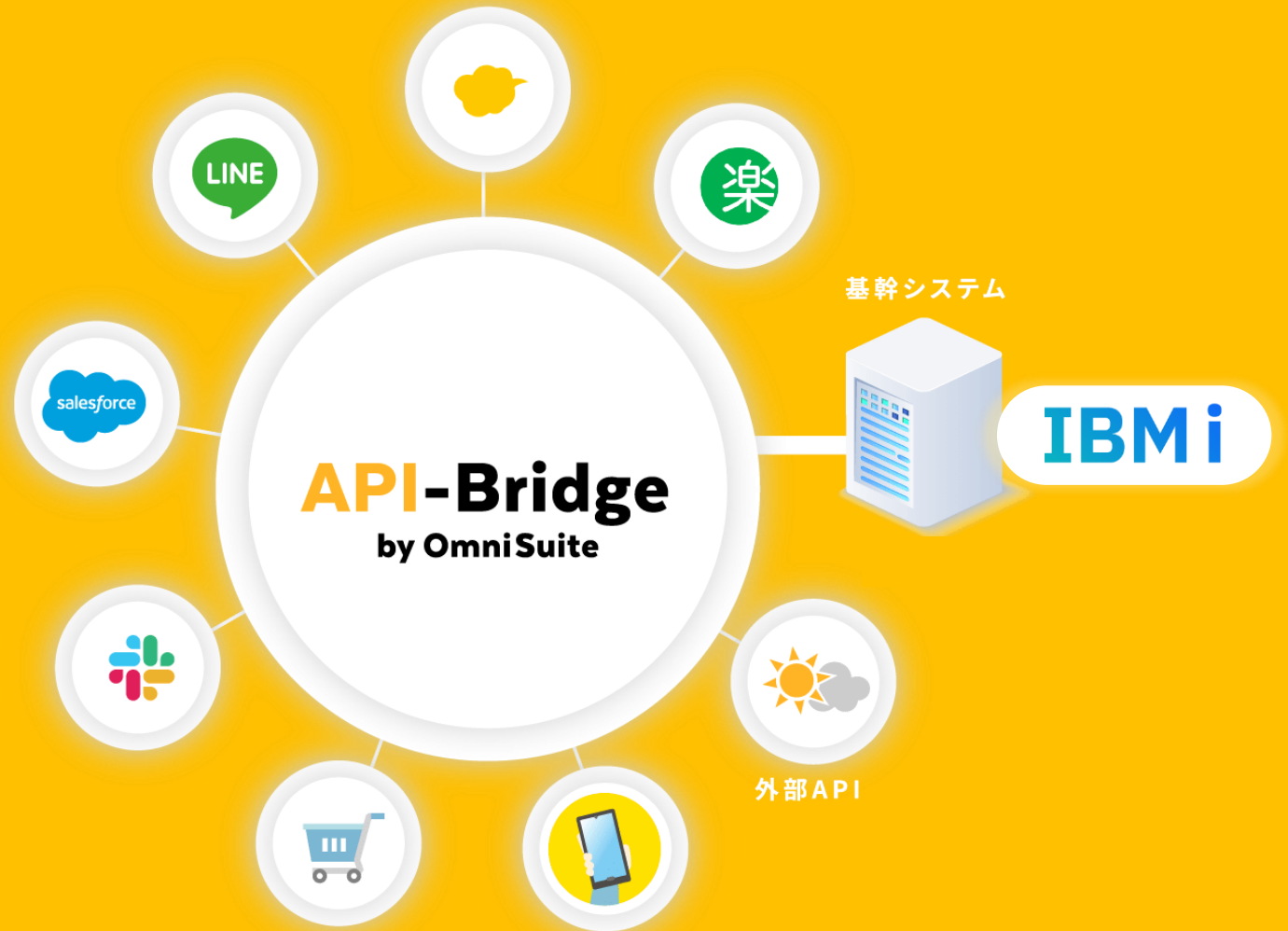
★

# API-Bridge

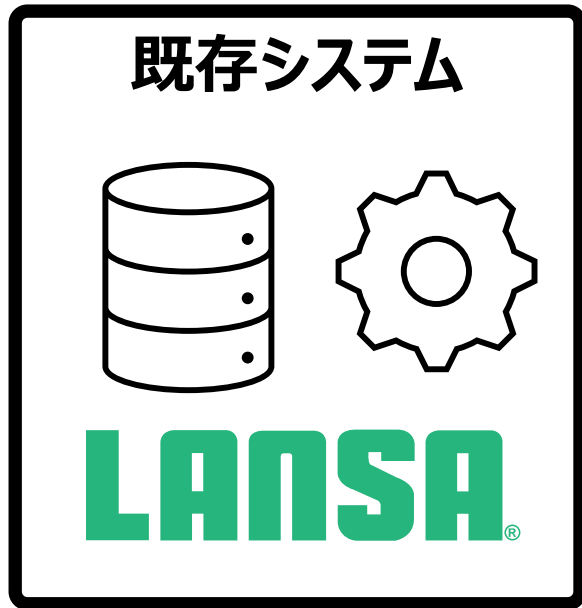
## Concept

コンセプト

「様々なクラウドサービスと、  
IBM i の基幹データをつなぐ架け橋」  
としてAPI-Bridgeは生まれました。



# Visual LANSA Connectors



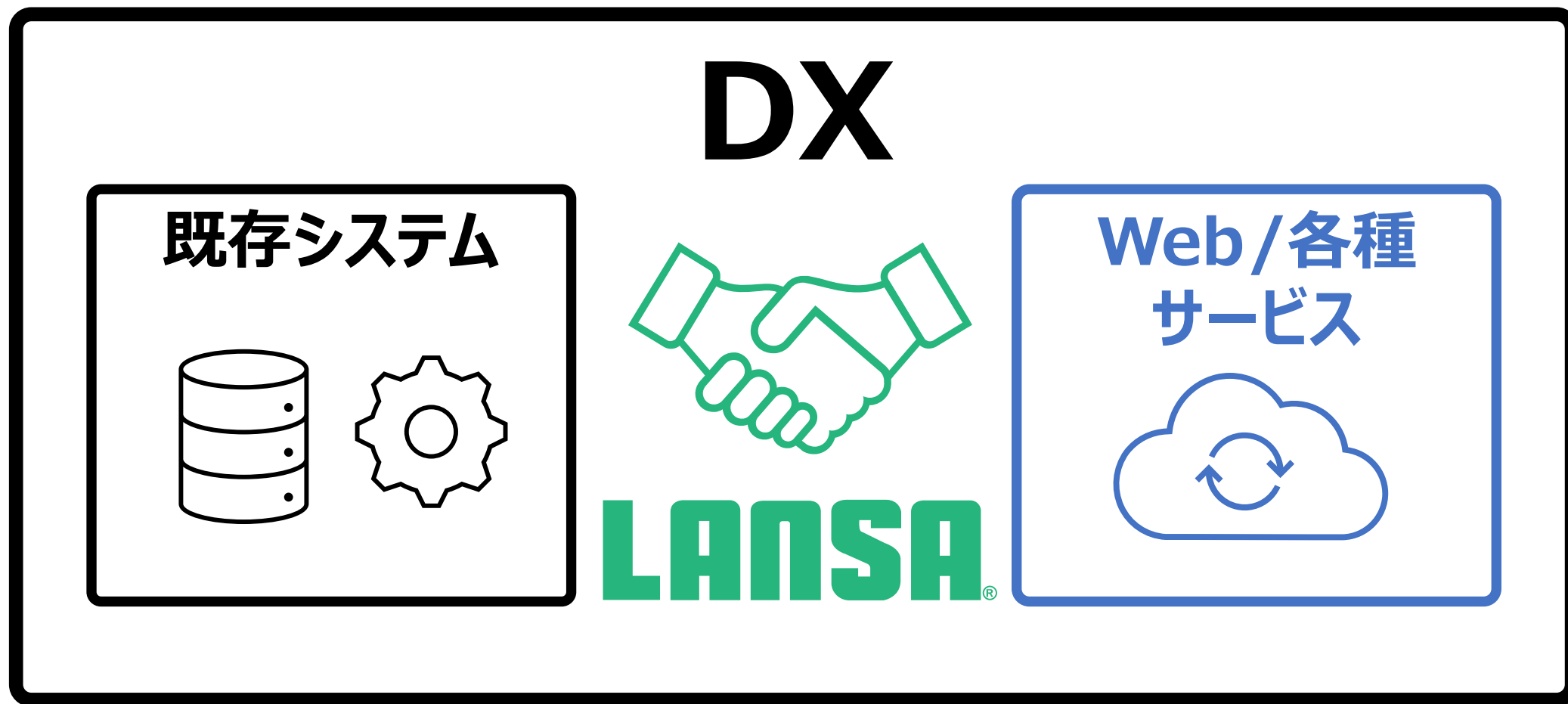
200超のサービス連携

**cd data**  
Visual LANSA  
Connectors





# 相互連携とDX推進



A network diagram consisting of several grey cylindrical nodes connected by thin teal lines. The nodes are arranged in a non-uniform pattern, with some having multiple connections, creating a web-like structure. The background is a dark blue gradient.

**LANSA**®は  
システムの継承とDXを  
お手伝いします