

【セッションNo. 2】

Delphi 10.2 Tokyoで実現する 広範な接続性

- APIで変わる情報システム

エンバカデロ・テクノロジーズ
日本法人代表

藤井 等 様

【アジェンダ】

- APIとは
- Delphi的見地からのAPI
 - APIを利用する
 - APIを公開する
- Webアプリケーションサポートの新しいかたち
- 10.2 Tokyoの広範なモダナイゼーションサポート

■ APIとは

- アプリケーション開発者向けにプラットフォームの機能を利用できるようにしたもの
- よく知られているAPIは...

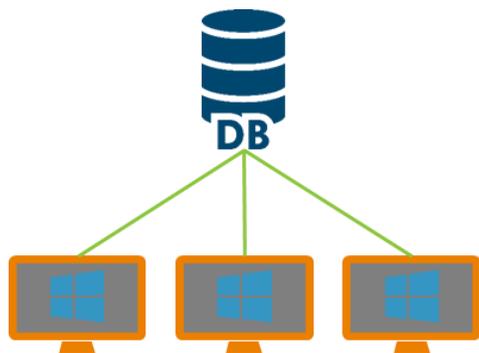
Windows API

- 一方、今話題のAPIは...



■ 従来のシステムは閉じた世界

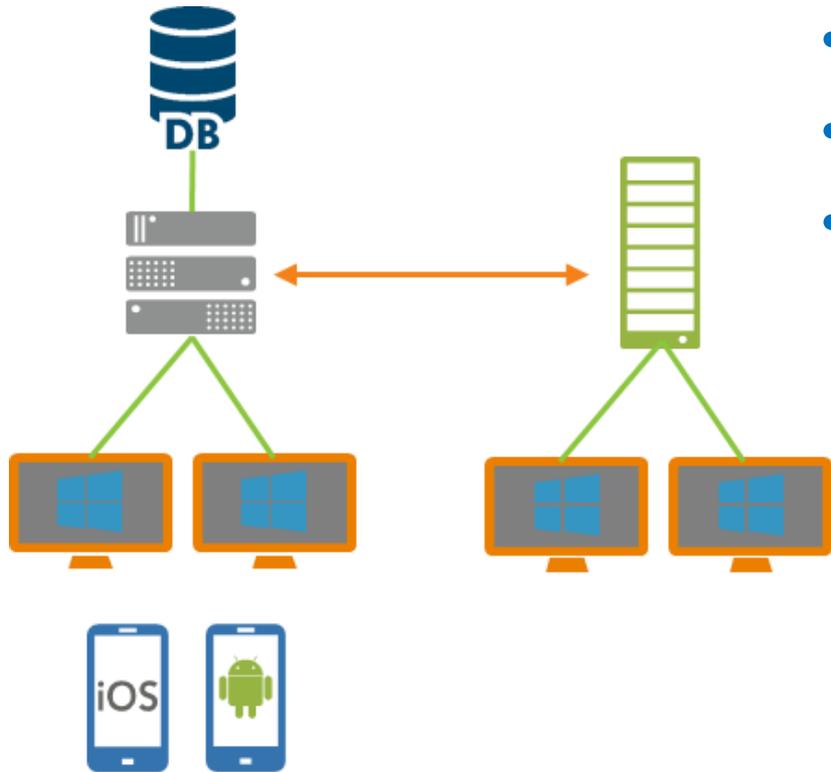
- クライアントサーバーシステムは独立した島のような



- 閉じたシステム
- データの連携は基本的にインポート/エクスポート
- 必要とするすべての機能をシステムに実装

■ 多層化でシステム間連携が可能に

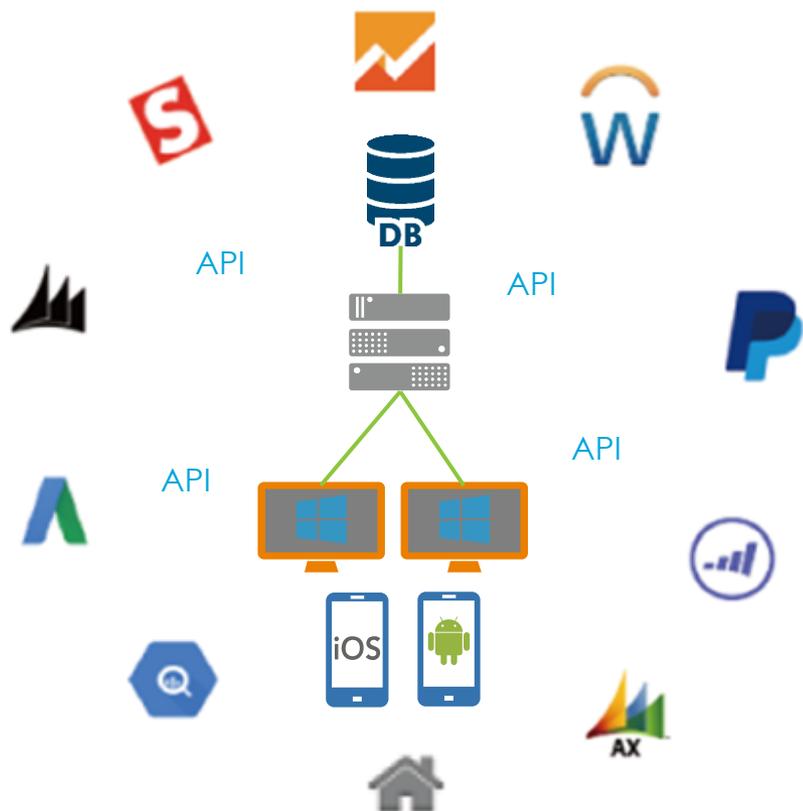
- システム同士が連携できるように、中間のサーバー機能が介在



- つながるシステム
- データの連携は相互通信
- 複数のシステムが連携することで全体の機能を提供

■ APIによって動的なシステムに

- クラウドサービスの使用により、そのとき必要とする機能を動的に利用する形態へと変化



- 動的なシステム
- データの連携はAPIが仲介
- 必要な機能をクラウドサービスを活用することで実現

■ Delphi的見地からのAPI

① APIを利用する

Delphiアプリケーションで外部のAPIを利用する

② APIを公開する

Delphiアプリケーションの機能を外部から利用できるようにする

■ 外部APIにはどのようなものがあるか

- SaaS + カスタムアプリ
- SOAP/REST
- XML/JSON
- SDK
- FileTransfer
- Auth (Basic / OAuth)



■ 例えば...



- Salesforce
 - クラウドベースのCRMシステムを提供

外部APIを使うと



SalesForceに登録された顧客情報を社内システムで表示



C/S型の従来システムで実行した発注処理の情報をSalesForceにも登録



外部システムを使って、SalesForceにリード情報を登録

■ 外部API利用におけるチャレンジ

- 多くの開発者にとってなじみがあるのはデータベースとそれを扱うSQL
- Web APIの習得自体が大きな負担
- クラウドサービスごとに異なる仕様
- 開発言語に依存したSDK
- 頻繁に変わるAPIとそれに対応するためのメンテナンスコスト

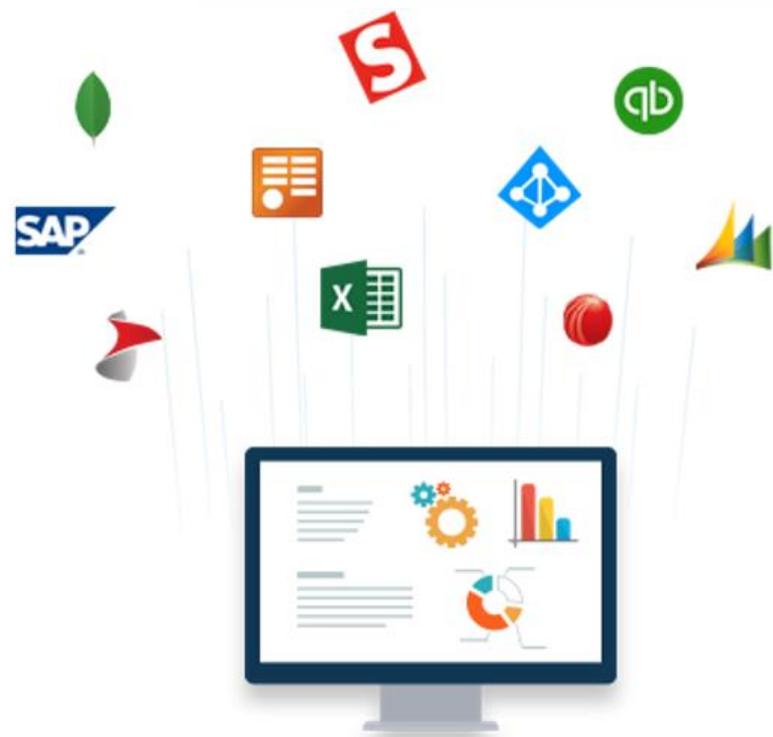


■ そこで



• Enterprise Connectors

- エンタープライズデータソースへのアクセス性を提供するコンポーネント



Delphiの標準データアクセス
フレームワークFireDACにより
多様なエンタープライズデータ
／クラウドサービスに接続可能

- コンポーネントによる共通アクセス
- TDataSet、TFieldなどのデータ型をそのまま利用可能
- DBコントロールで表示、編集
- データエクスプローラ、フィールドエディタなどを利用可能

■ Enterprise Connectorsのしくみ

- コンポーネントによる共通アクセスを実現

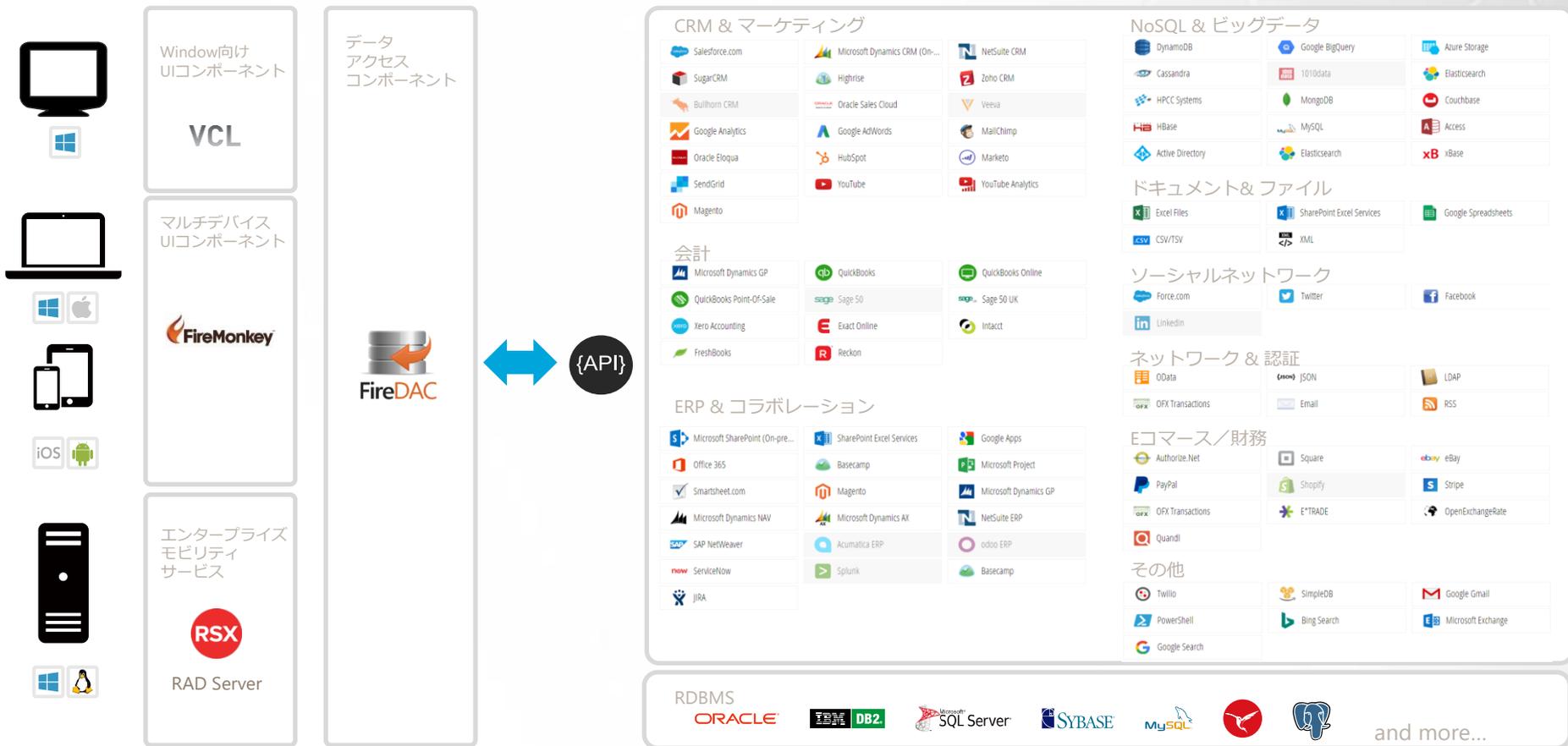


Salesforce & Force.com	Microsoft Dynamics CRM	Microsoft Dynamics GP
Microsoft Dynamics NAV	Microsoft Dynamics AX	MongoDB
Microsoft SharePoint	Apache Cassandra	Apache HBase
Apache Spark SQL	NetSuite CRM & ERP	Sage 50 Accounting
Sage 50 UK	Xero Accounting	Exact Online
FreshBooks Accounting	SAP NetWeaver	OData Services
JSON Services	Microsoft Excel	SharePoint Excel Servi...
Twitter	Facebook	LDAP Directory Services
Amazon DynamoDB	Amazon SimpleDB	Google Analytics

```
select * from Account where  
Industry = "Healthcare"
```

Enterprise Connectorsのしくみ

- FireDACによりRDBMSと同じ感覚で操作可能に



Enterprise Connectors

FireDACコンポーネントでクラウドデータにアクセス

クラウドデータをSQLで参照

The screenshot shows the RAD Studio 10.2 interface. The main window is titled "EntConn_Twitter - RAD Studio 10.2 - Unit1". The menu bar includes "ファイル", "編集", "検索", "表示", "リアクティング", "プロジェクト", "実行", "コンポーネント", "ツール", and "ヘルプ". The toolbar contains various icons for file operations, execution, and development. The "WelcomePage" and "Unit1" tabs are visible. A "FireDAC クエリ エディタ - [TweetsTable]" window is open, displaying the following SQL command:

```
SELECT Created_At, Favorite_Count, Retweet_Count, Text
FROM Tweets
Where SearchTerms='Delphi/400'
Order by Created_At
Limit 100;
```

Buttons for "実行(E)", "次のレコードセット(N)", and "更新 SQL エディタ(U)" are present. A checkbox for "自動ロールバック" is also visible. Below the SQL editor, the "RecordSet" tab is active, showing a table with the following data:

Created_At	Favorite_Count	Retweet_Count	Text
2017/10/12 13:53:07	0	0	[400 Acessos] Curso Básico e Rápido de Borl...
2017/10/12 13:53:11	0	0	[400 Acessos] Curso Básico e Rápido de Borl...
2017/10/14 6:06:28	0	0	@kathe_Delphi Tipo 8, necesito ir a comprar
2017/10/19 11:24:06	0	0	IBM 向けビジュアル開発ツールDelphi/400が新
2017/10/19 15:54:05	0	0	IBM 向けビジュアル開発ツールDelphi/400が新
2017/10/19 16:21:05	0	0	ミガロ.が提供するIBM 向けビジュアル統合開発

On the right, the "データ エクスプローラ" (Data Explorer) shows a tree view of the "CData.Twitter.Tweets" dataset with fields like "インデックス", "主キー", "外部キー", "ID", "IDLong", "SearchTerms", "Created_At", "Text", "Lang", "Source", "Favorited", "Favorite_Count", "Retweeted", and "Retweet_Count". The "ツール パレット" (Tool Palette) at the bottom right shows categories like "Standard", "Additional", "Win32", "System", and "Win 3.1".

Enterprise Connectors

EntConn_Twitter - RAD Studio 10.2 - Unit1 [ビルド完了]

検索 表示 リファクタリング プロジェクト 実行 コンポーネント ツール ウィンドウ ヘルプ

32ビット Windows

WelcomePage Unit1

データ エクスプローラ

CDData.Twitter.Tweets

- インデックス
- 主キー
- 外部キー
- ID
- IDLong
- SearchTerms
- Created_At
- Text
- Lang
- Source
- Favorited
- Favorite_Count
- Retweeted

オブジェクト インспекタ

ReversegeosearchProc TFD

プロパティ イベント

検索

DBコントロールにクラウドデータを表示

requestsProcReversegeosearchProc

FDConn_Twitte

FDQuery_Insert

FDQuery_Select

DataSource1

Standard

Additional

Win32

System

Win 3.1

Dialogs

Data Access

Data Controls

dbExpress

Datasnap Server

Xml

LiveBindings

LiveBindings Misc

Net

193: 1 挿入 変更あり コード デザイン 履歴

■ APIの公開とは？

- Delphiアプリケーションの機能を外部*から利用できるようにする

* 社内の別システムも含む



■ DelphiでAPIを公開するには

- 中間サーバー機能を使います

手法1 : RAD Serverを使う

手法2 : DataSnapを使う



- DelphiでREST APIを構築／公開するための中間サーバー機能を用意
- データアクセスの最適化、モバイルクライアントの管理機能、認証などの機能を用意



- 中間サーバー構築のための実装技術を提供
- SDKによりすべての機能をプログラミング可能

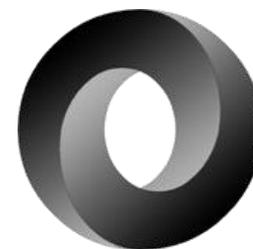
開発工数	小	中
管理機能	あり（ユーザー／グループ管理、認証、分析など）	なし（要実装）
モバイル対応	あり（デバイス種別の認識、通知サービスなど）	なし（要実装）
配布の手間	小	中
ライセンス費用	別途必要	開発ライセンスに含まれる

■ APIはRESTの仕様で実装

- REST (REpresentational State Transfer)
 - ネットワーク上のデータ (リソース) を一意なURIで表す設計手法
- データに対するSQL操作 (INSERT, SELECT, UPDATE, DELETE) は、HTTPメソッドで表現する
- URI+HTTPメソッドが完全に同一なリクエストは常に等価な結果が得られる

処理	HTTPメソッド	URL	対応するSQL操作
登録	POST	/resource/	INSERT
取得	GET		SELECT
更新	PUT	/resource/{item}	UPDATE
削除	DELETE		DELETE

■ やり取りするデータ形式

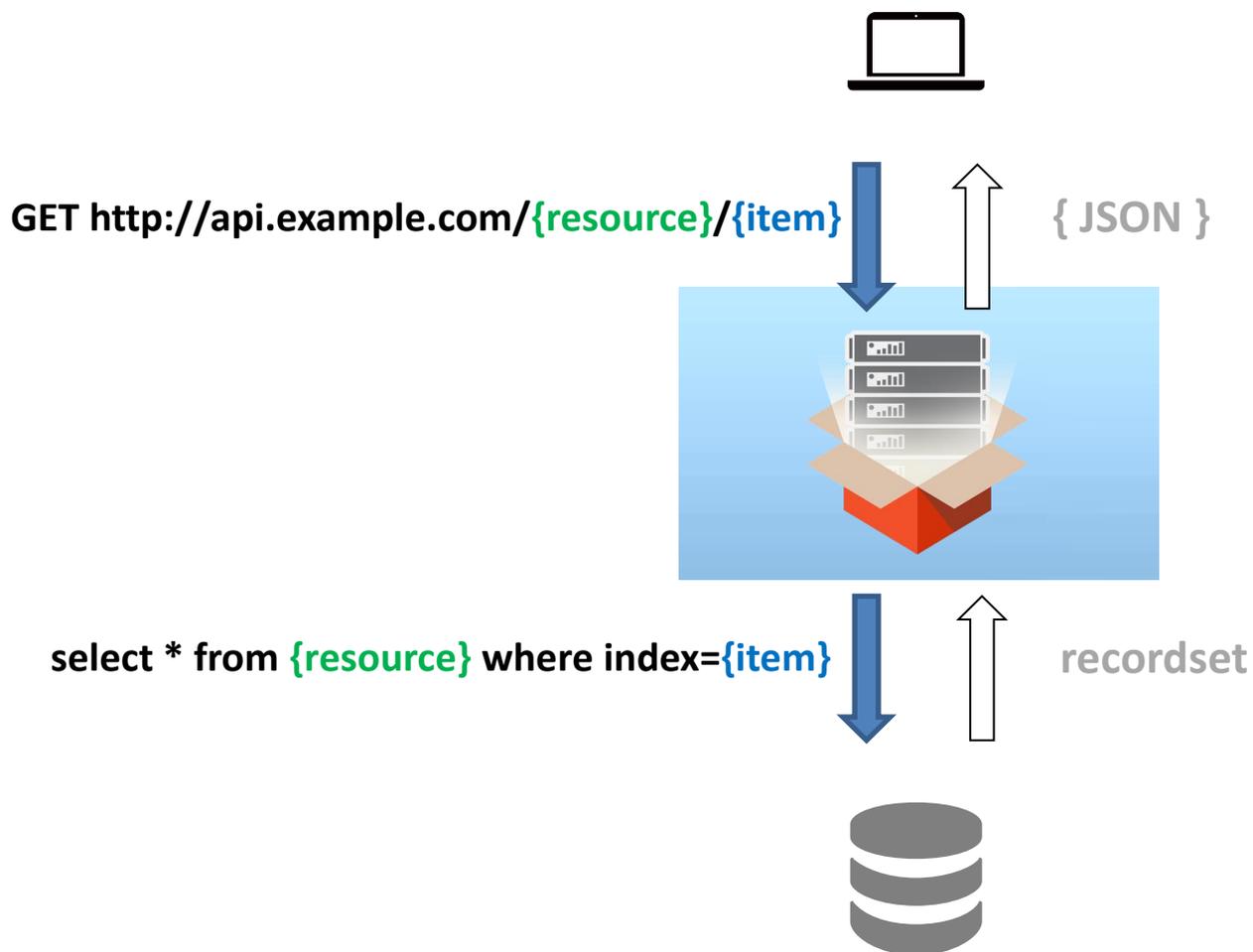


- JSON (JavaScript Object Notation)
 - 言語非依存の軽量のデータ交換形式
 - 現在多くのWeb APIがJSONによってデータを交換
- DelphiではJSONデータ型を扱うフレームワークを用意
 - TJSONObject、TJSONArray、TJSONNumber、TJSONString...
 - TJsonWriter、TJsonReader...

```
[{"id":2747,"type":"Song","title":"Lady Madonna","artist":
{"id":19,"type":"Artist","nameWithoutThePrefix":"Beatles","useThePrefix":true,"name":"The
Beatles"},"chordsPresent":true,"tabTypes":["PLAYER","TEXT_GUITAR_TAB","CHORDS","TEXT_BASS_TAB"]},
{"id":59530,"type":"Song","title":"Lady Madonna","artist":{"id":1904,"type":"Artist","nameWithoutThePrefix":"Marcel
Dadi","useThePrefix":false,"name":"Marcel Dadi"},"chordsPresent":false,"tabTypes":["PLAYER"]},
{"id":12927,"type":"Song","title":"Papa Don't Preach","artist":
{"id":2959,"type":"Artist","nameWithoutThePrefix":"Madonna","useThePrefix":false,"name":"Madonna"},"chordsPresent":tr
ue,"tabTypes":["PLAYER","CHORDS"]}, {"id":10767,"type":"Song","title":"Like a Virgin","artist":
{"id":2959,"type":"Artist","nameWithoutThePrefix":"Madonna","useThePrefix":false,"name":"Madonna"},"chordsPresent":tr
ue,"tabTypes":["PLAYER","TEXT_GUITAR_TAB","CHORDS","TEXT_BASS_TAB"]}, {"id":22862,"type":"Song","title":"La Isla
Bonita","artist":
{"id":2959,"type":"Artist","nameWithoutThePrefix":"Madonna","useThePrefix":false,"name":"Madonna"},"chordsPresent":tr
ue,"tabTypes":["PLAYER","CHORDS"]}, {"id":22864,"type":"Song","title":"Material Girl","artist":
```

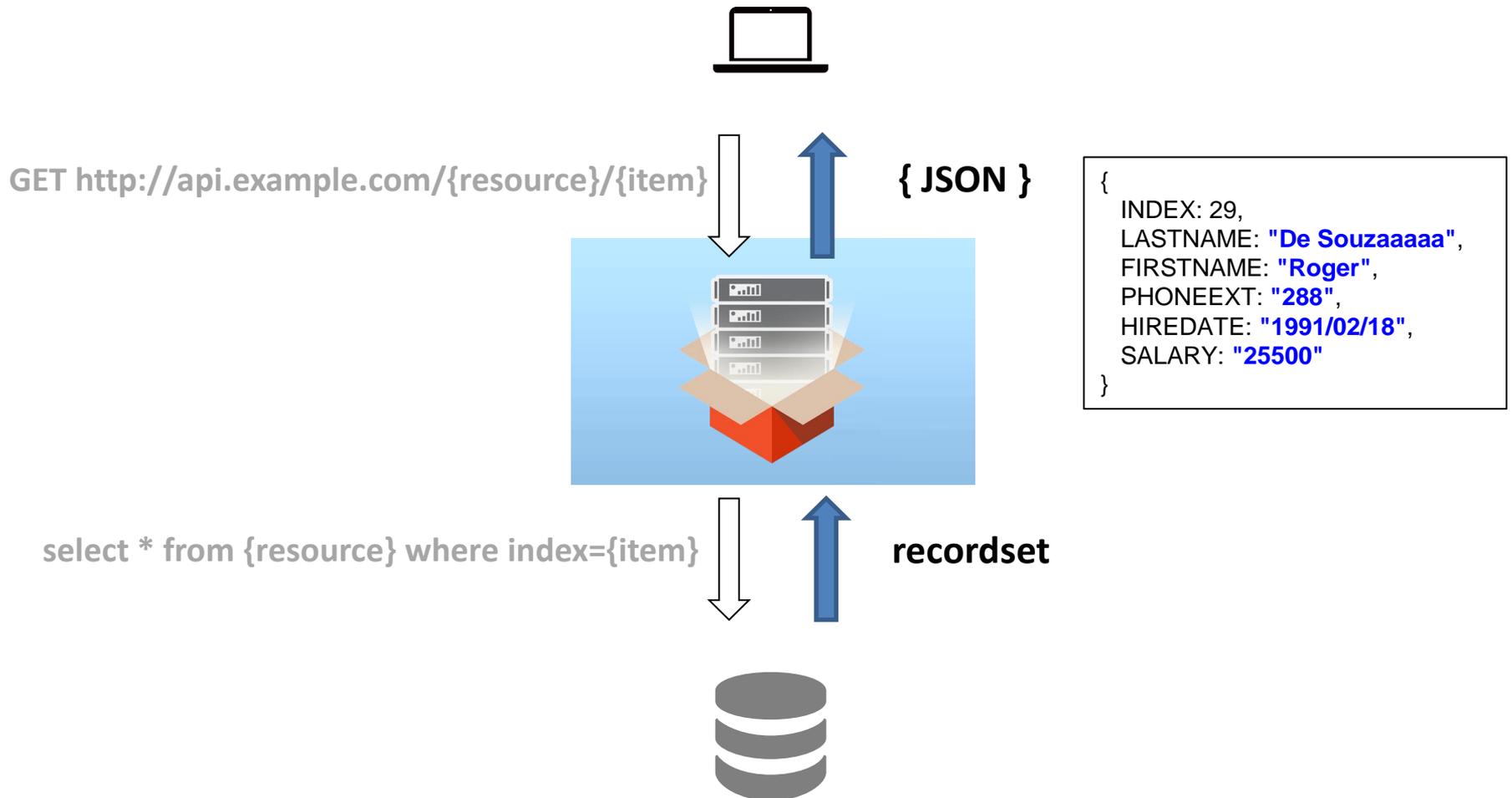
■ RESTのデータフロー例（リクエスト送信）

- 中間サーバがリクエストに等価なクエリをDBに投げる



■ RESTのデータフロー例（応答の受信）

- DBからのデータはJSONに変換してクライアントに戻す



API公開のイメージ



デスクトップ



モバイル



他システム



中間サーバー

クラウドサービス

Salesforce & Force.com	Microsoft Dynamics CRM	Microsoft Dynamics GP
Microsoft Dynamics NAV	Microsoft Dynamics AX	Microsoft SharePoint
Netsuite CRM & ERP	QuickBooks Desktop	QuickBooks Online
QuickBooks Point of Sale	Sage US	Sage 50 UK
Xero Accounting	Exact Online	Intacct
FreshBooks Accounting	SAP NetWeaver	OData Services
JSON Services	Microsoft Excel	SharePoint Excel Services
Twitter	Facebook	LDAP Directory Services
Amazon DynamoDB	Amazon SimpleDB	Google Analytics
Google Apps	Google Apps	Google Sheets
Office 365	Gmail	OFX Financial Accounts
PowerShell	Email	RSS Feeds
MongoDB	Google BigQuery	Azure Table
Apache Cassandra	Couchbase Server	Apache HBase



基幹システム



RDBMS

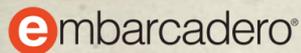


and more...

■ Webアプリケーションサポートの新しいかたち

• Sencha

- Webアプリケーション構築ソリューション
- モダンなWebアプリケーションを効率的に構築



エンバカデロ親会社のアイデラが、Senchaを買収 エンバカデロへの統合により開発ツールビジネスを強化

2017年8月29日
エンバカデロ・テクノロジーズ

エンバカデロ・テクノロジーズ（本社：米国テキサス州オースティン、日本法人：東京都文京区、日本法人代表：藤井 等、以下エンバカデロ）の親会社であり、データベース、開発ツール、テスト管理ツールのリーディングプロバイダーであるアイデラ（Idera Inc. 米国テキサス州ヒューストン）は、主要なWebアプリケーション開発プラットフォームであるSenchaの買収に合意したことを発表しました。なお、Sencha（米国カリフォルニア州シリコンバレー）は、エンバカデロに統合されます。

SenchaのWebアプリケーション開発プラットフォームは、企業がデスクトップ、タブレット、スマートフォン向けの機能豊富なクロスプラットフォームWebアプリケーションのシームレスな設計、開発、テストを可能にします。最近の調査では、Senchaのフルスタックエンタープライズソリューションを使用している顧客は、他のWebアプリケーションフレームワークを使用している場合よりも、最大で50%も生産性が向上しています。

■ Senchaとは

- エンタープライズWebアプリケーションを迅速に構築

Ext JS

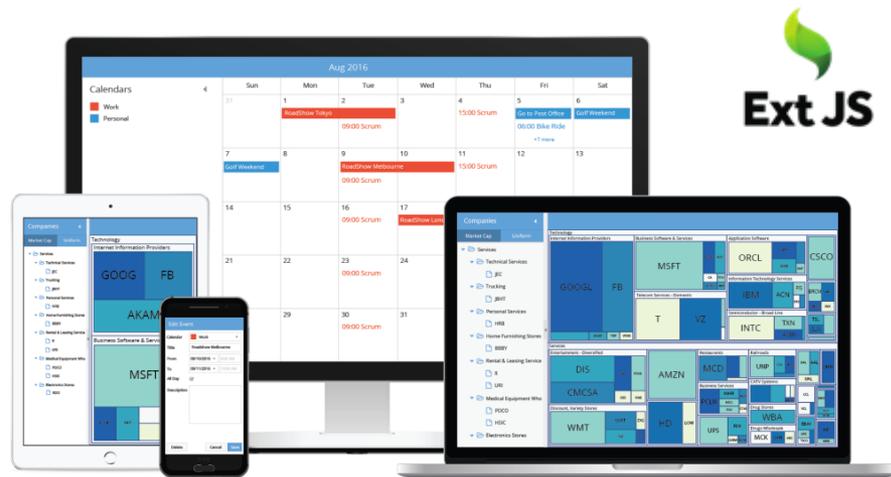
- JavaScriptによるクロスプラットフォームアプリ開発
- カスタマイズ可能なUI部品
- 幅広いブラウザをサポート

Sencha Architect

- ドラッグ&ドロップによるHTML5アプリケーション構築をサポート

Sencha Test

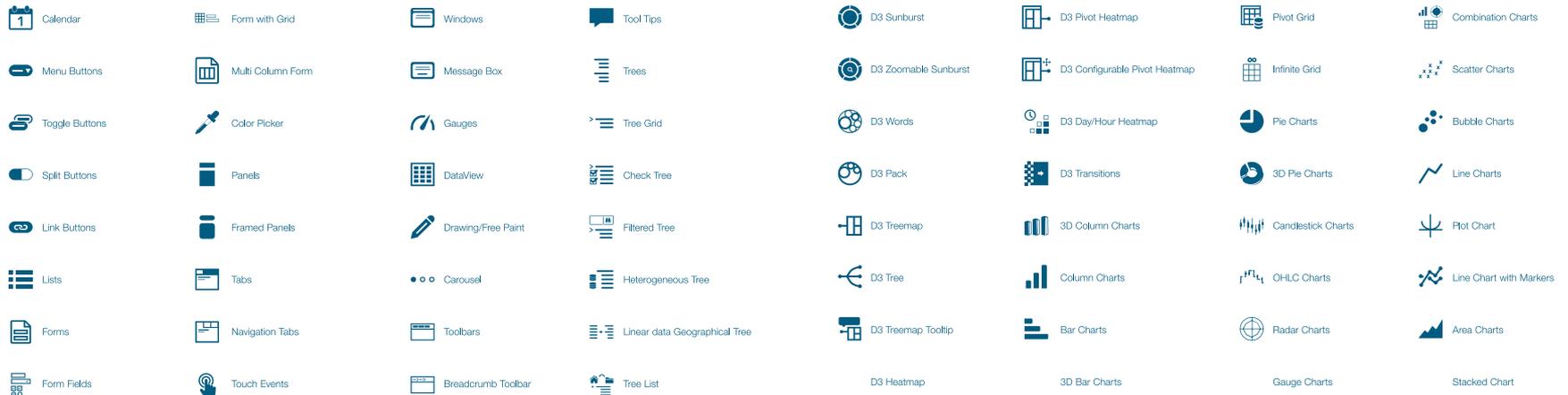
- Ext JSアプリの単体・機能テストの最善のソリューション



■ 豊富なコンポーネントによる開発

「Senchaを利用することで、1つのUI部品の作成に100時間以上の節約ができました。UI部品は開発だけではなく長期的な保守が必要なことを忘れないで下さい。Senchaは測定できないほど優れた価値を持っています」

米国の大手金融サービス企業の開発マネージャー



SenchaでAPIを利用する

```

1 Ext.define('MyApp.model.Customer', {
2   extend: 'Ext.data.Model',
3   requires: [
4     'Ext.data.Field.Field'
5   ],
6   fields: [
7     {
8       mapping: 'Original.CUSINO',
9       name: 'CUSINO'
10    },
11    {
12     mapping: 'Original.COMPANY',
13     name: 'COMPANY'
14    },
15    {
16     mapping: 'Original.ADDR1',
17     name: 'ADDR1'
18    },
19    {
20     mapping: 'Original.ADDR2',
21     name: 'ADDR2'
22    },
23    {
24     mapping: 'Original.CITY',
25     name: 'CITY'
26    }
27  ],
28  });

```

レコードのスキーマを定義

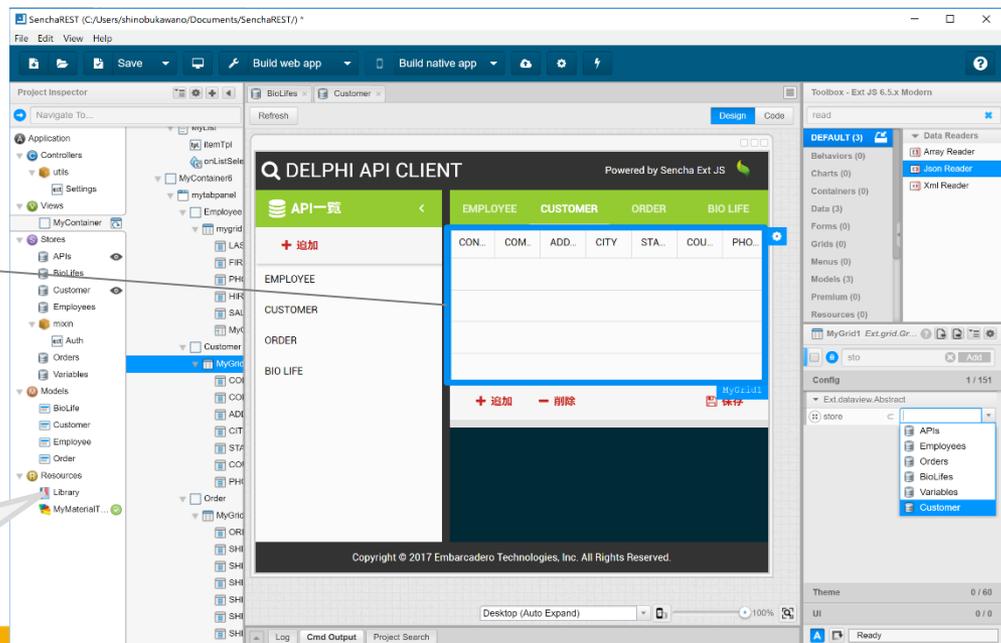
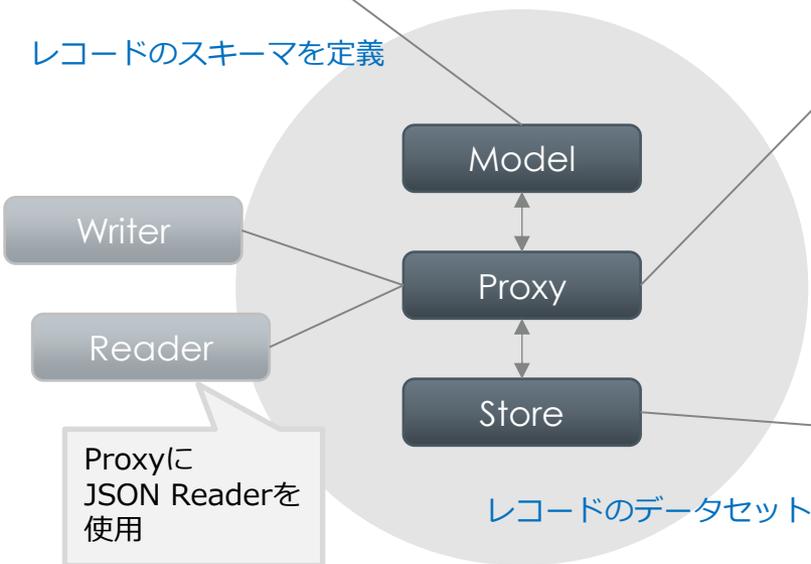
```

1 Ext.define('MyApp.store.Customers', {
2   extend: 'Ext.data.Store',
3   mixins: {
4     auth: 'MyApp.store.mixin.Auth'
5   },
6   requires: [
7     'MyApp.model.Customer',
8     'Ext.data.proxy.Rest'
9   ],
10  constructor: function(cfg) {
11    var me = this;
12    cfg = cfg || {};
13    me.callParent([Ext.apply({
14      storeId: 'Customers',
15      model: 'MyApp.model.Customer',
16      proxy: {
17        type: 'rest',
18        url: 'http://10.120.0.145/dbdemos/customer'
19      },
20      cfg
21    })]);
22  }
23  });

```

ProxyのURLに
RAD Serverの
エンドポイント
を指定

データソースを指定



Storeと
Gridを関連付け

10.2 Tokyoの広範なモダナイゼーションサポート

