A decorative graphic on the left side of the slide features a network of colored lines (yellow, blue, green, pink) with circular nodes. These lines connect to various icons: a Windows logo at the top, an Android robot in the middle, a Bluetooth symbol at the bottom, and an Apple logo at the very bottom left.

30TH EMBARCADERO DEVELOPER CAMP

第30回 エンバカデロ・デベロッパーキャンプ・ツアー

【E3】Delphi/C++テクニカルセッション

「移行プロジェクトでもモバイル開発でも必須！
今からでも間に合うデータモジュール活用法」

株式会社ドリームハイブ
代表取締役 山本悟

The logo for Embarcadero, featuring a red circle with a white 'e' inside, followed by the word 'embarcadero' in a bold, lowercase, sans-serif font.

自己紹介

名前: 山本 悟 (やまもと さとる)

- 代表取締役 & ITコンサルタント

会社: 株式会社 ドリームハイブ

- 会社URL : <http://www.dreamhive.co.jp/>
- 山本のブログ : <http://blog.dreamhive.co.jp/yama/>
- facebook : <http://www.facebook.com/kryu2>
- twitter : <https://twitter.com/kryu2>

山本はこんな感じの人:

- 17歳からIT業界へ
- Delphi は1.0からの親友
- テレビ埼玉に出たり、リクルート系雑誌に載ったり
- ドリームハイブの経営、ITコンサルティング、システム開発、スピーカーなどが今の主な仕事



アジェンダ

- はじめに
 - データモジュールとは
 - データモジュールを使っていないとどうなる？
 - 使った際の利点
- データモジュール再入門
 - はじめての人向け
 - やっちゃった人向け
 - その他TIPS
- デモデモでも～
- まとめ

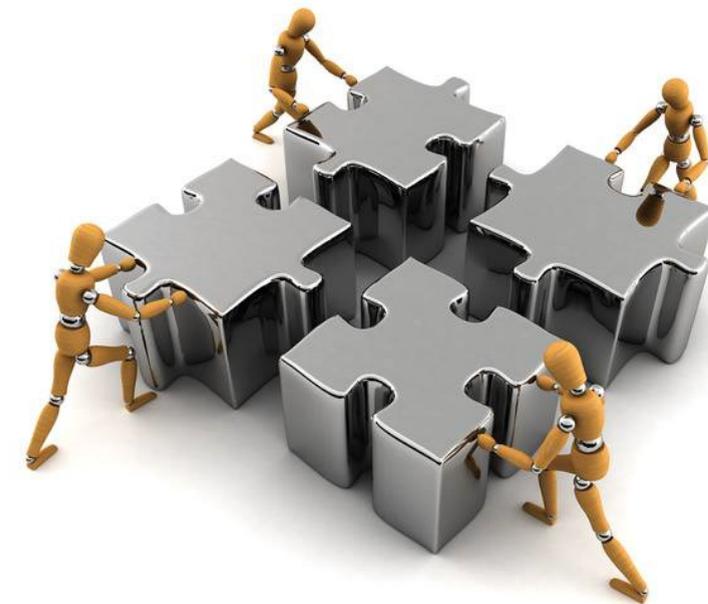
移行プロジェクトでもモバイル開発でも必須！
今からでも間に合うデータモジュール活用法

はじめに

データモジュール使っていますか？

データモジュールとは

- データモジュールは、
非ビジュアルコンポーネントのコンテナで、
たとえば、データアクセスコンポーネントを
隠蔽・集中管理することができます。



使っていないと...

使わなくてもアプリケーションは開発できますが
フォーム上にポトペタしていると

- UIの変更に巻き込まれる
 - システム拡張
 - 設計変更(データベースの設計変更も含む)
- などがあると死にます。



問題が起こるケース

- それは**要求変更**！！
- 例)
 - 複数のPCから見られるのは当たり前だろオーダーが入った！
 - 社長からWebが主流らしいじゃんの鶴の一声が出た！
 - 最近流行のモバイルアプリに対応しろオーダーが入った！

何が問題となるのか？

- 複数クライアントから**同時アクセス**を想定しなければならない
- クライアント**環境が複数**存在する事になる
- クライアントへの**配布**が難しい
- クライアントアプリケーションの**バージョン管理**が難しい
- 他のシステムやライブラリを利用する確率が増える
- 社内サーバに社外から直接アクセスできなければ、**中継サーバ**が必要になる
- 社内外間の**ネットワーク負荷**が増大する
- DBへの**同時接続数**の問題が発生する

大切なことなので何回も言います

思い出しましょう。

データモジュールは、
非ビジュアルコンポーネントのコンテナで、
たとえば、データアクセスコンポーネントを
隠蔽・集中管理することができます。



→ コレを使うと UIとBLを切り分けることができるのです。

UI/BLの分離の利点

- テストの範囲が狭くなる
 - ユニットテストが変更された箇所のみで良くなり、テスト工数が減る
 - UI
 - データモジュール
- マルチデバイス開発がしやすい
 - 各デバイス用UIが増えても、UIのみの処理に集中できる

UI/BLの分離の注意点

- 詰め込まない！
 - UIの操作処理をBL側に組み込んだりしない！
 - **共通化**できるところとできないところの切り分けはしっかりやろう！
 - **in/outはシンプル**に！
- コンポーネントを貼りまくらない！
 - データベースのテーブル毎にコンポーネントを全部置いて、しかも最初に全部オープンすると遅い！
- TXXQueryを使う！
 - TXXTableを使うと全データを参照するので遅い！
必要なデータのみにアクセスすること。

使っているとこんな風に幸せになれます

- UIの変更のときも
- システム拡張のときも
- 設計変更(データベース側も含む)のときも

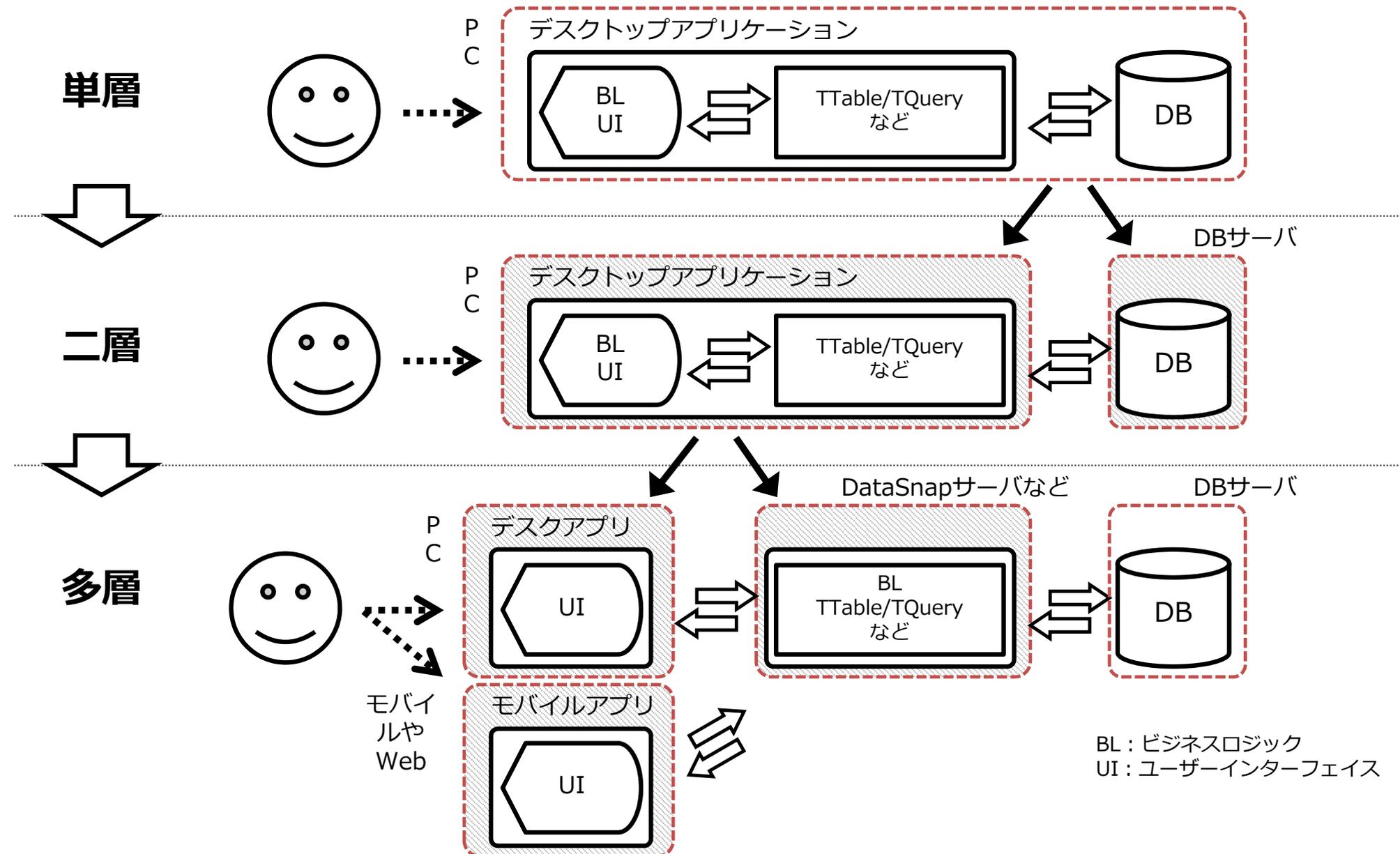
最初からデータモジュールを使っておけば、
少ないダメージですみます。

UIとBLを分離していると素敵なのです。

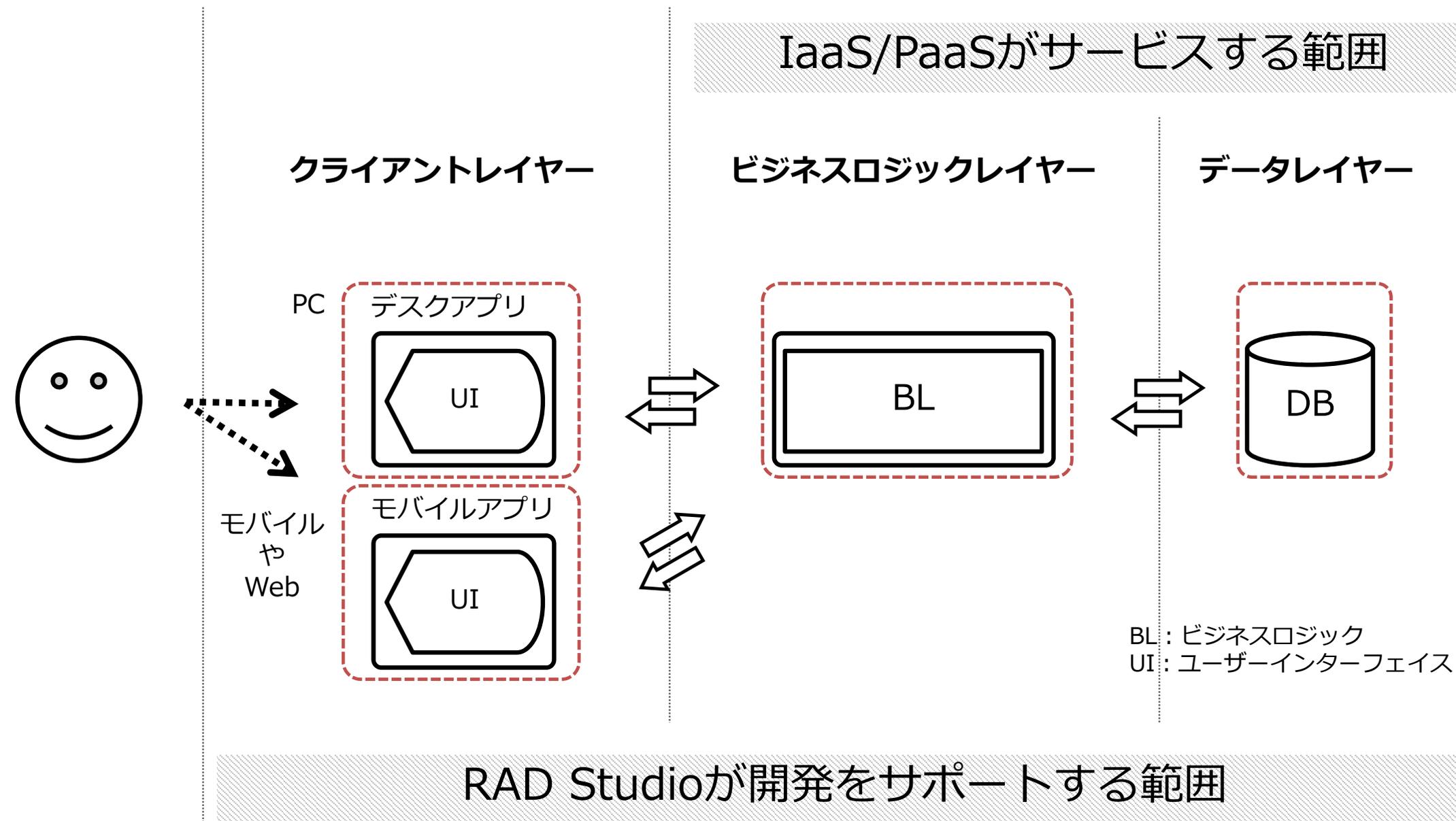
動作イメージを掴んでみましょう

データモジュールによる拡張しづらさの緩和

- 1層から
2層、
3層へ



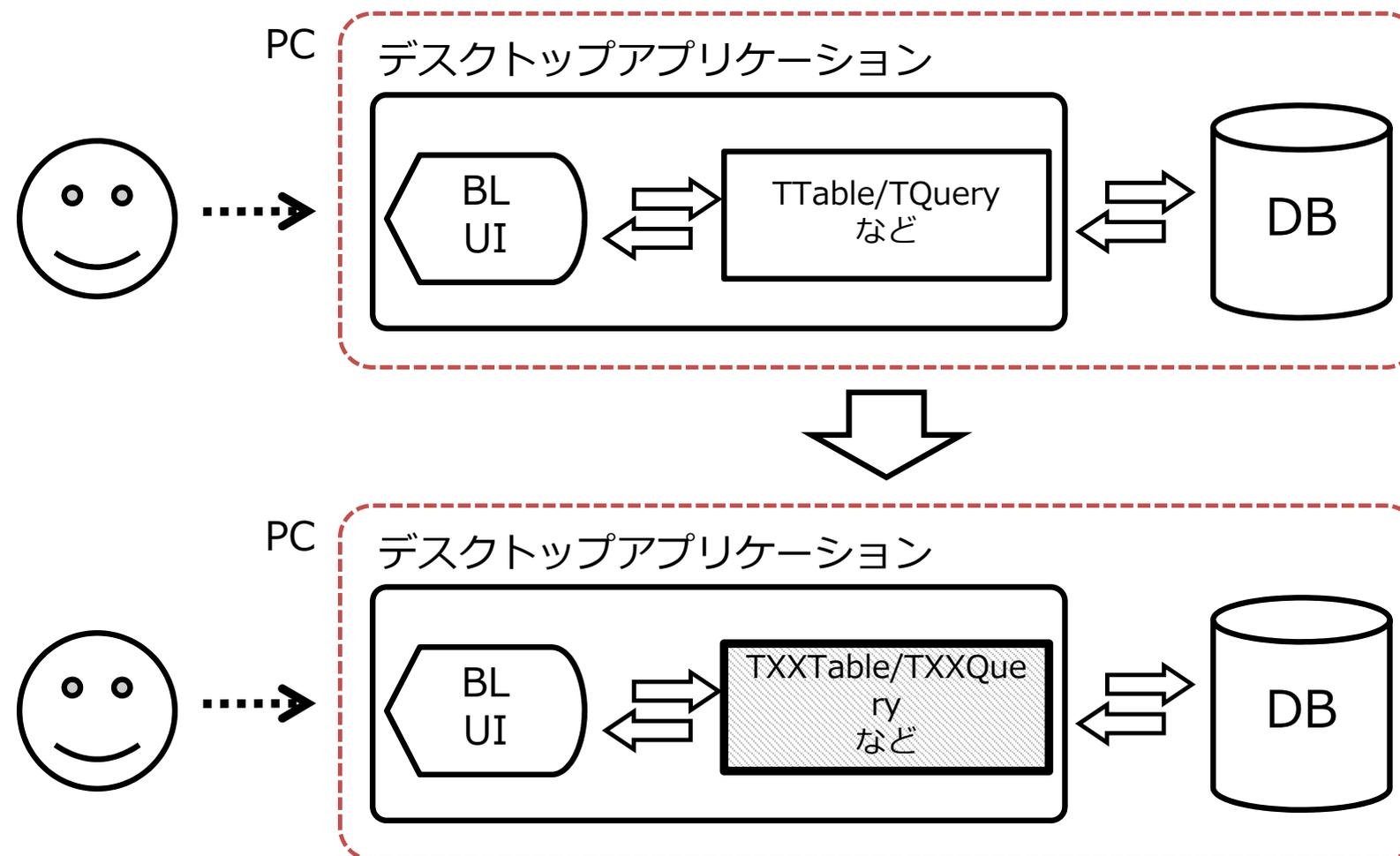
データモジュールによる拡張しづらさの緩和



データモジュールによる拡張しづらさの緩和

- 例えば、BDE などから FireDAC へ...

BL : ビジネスロジック
UI : ユーザーインターフェイス

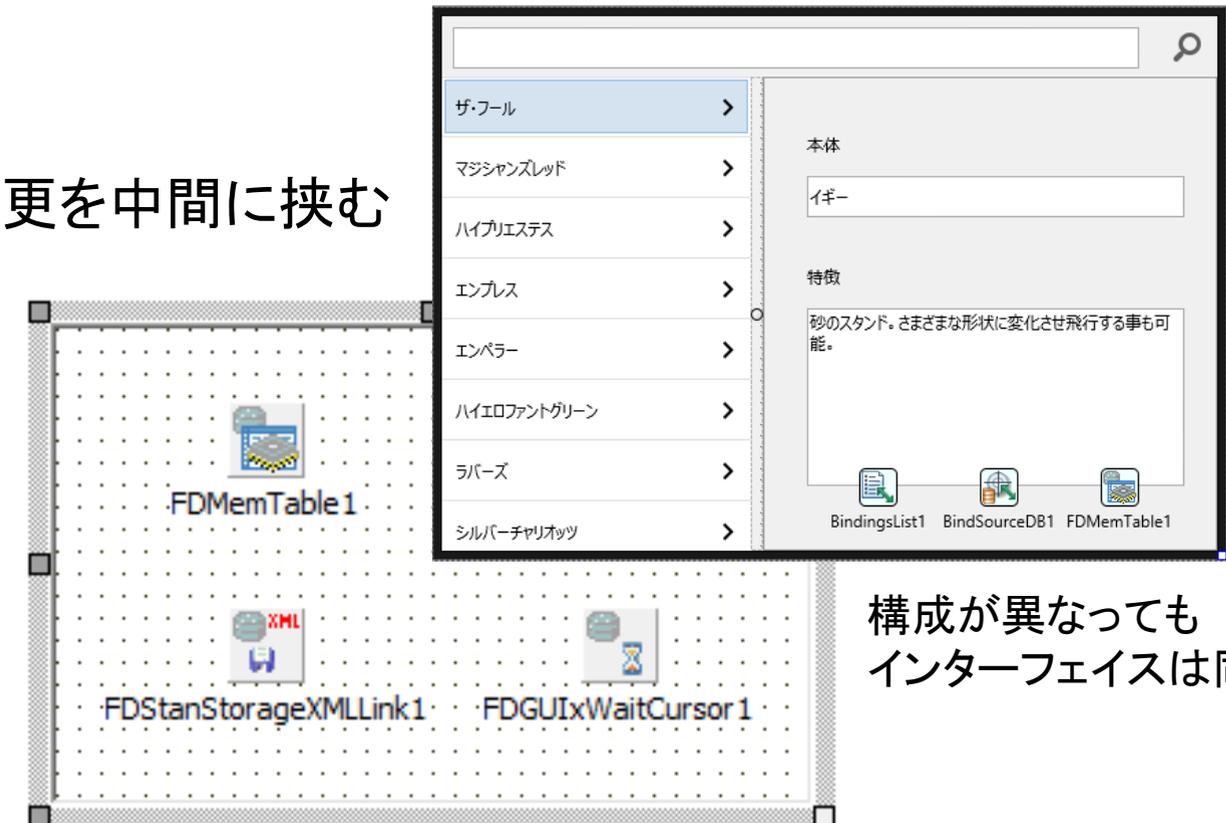


拡張しやすい構成の例

- 拡張しやすいとは
 - 理想は、変更数ゼロ、コードの追加のみで機能追加などができること
- 設計の例
 - データモジュールを利用する
 - クラスだけでなく、例外処理などもカプセル化する
 - データアクセスコンポーネントのアーキテクチャ変更を中間に挟む
 - メソッド内でオブジェクト参照を付け加える

```
var
  MyMemo: TMemo;
begin
  MyMemo := Memo1;

  MyMemo.Lines.Add('AAA');
  MyMemo.Lines.Add('BBB');
  MyMemo.Lines.Add('CCC');
end;
```



構成が異なっても
インターフェイスは同じ

工数を比較してみる

	規模	単層/二層	多層	
設計	小	😊😊😊	😊😊😊😐😐	多層は設計に 時間がかかる
	大	😊😊😊😐😐	😊😊😊😐😐	
製造	小	😊😊😊	😊😊😊😐😐	多層は製造にも 時間がかかる
	大	😊😊😊😐😐	😊😊😊😐😐	
テスト	小	😊😊😊	😊😊😊	多層はテストが楽
	大	😊😊😊😐😐	😊😊😊😐	
移行	小	😊😊😊	😊😊😊	多層は移行が楽
	大	😊😊😊😐😐😐😐	😊😊😊😐	
拡張	小	😊😊😊😐😐😐😐	😊😊😊	多層は 拡張しやすい
	大	😊😊😊😐😐😐😐	😊😊😊	

実はマルチデバイス対応にも便利なのです..

- VCL + Windows
- FireMonkey + Windows/Mac
- FireMonkey + Windows/Mac/iOS/Android

移行プロジェクトでもモバイル開発でも必須！
今からでも間に合うデータモジュール活用法

データモジュール再入門

作り方・直し方・注意点

さて、そんなデータモジュール。
どんな風に作って使うかをお見せします。

初めての方→新規に作るのを見てください

やっちゃった方→新規に作ったものからデータモジュールを
使ったものへの変更方法をお見せします

– フォームにがんがん貼っちゃっている人でも遅くない！

参考: テーブルの内容

今回のデモで使うデータは
テーブル一つだけです。

右の図はInterBaseの内容ですが、
同じデータをXML形式でも
用意しています。



XML

同じデータ
=



InterBase

Interactive SQL - 127.0.0.1/gds_db:C:\Users\Satoru\Desktop\DataModuleSampleFMX\JODATA.IB

ファイル(F) データベース(D) 編集(E) SQL関数(S) キーワード(K) クエリ(Q) ストアド プロシージャ(Z) トランザクション(T) ウィンドウ(W) ヘルプ(H)

テーブルおよびビュー

JOTABLE

フィールド

ST_ID
ST_NAME
RL_NAME
PART
INFO

```
SELECT * FROM JOTABLE;
```

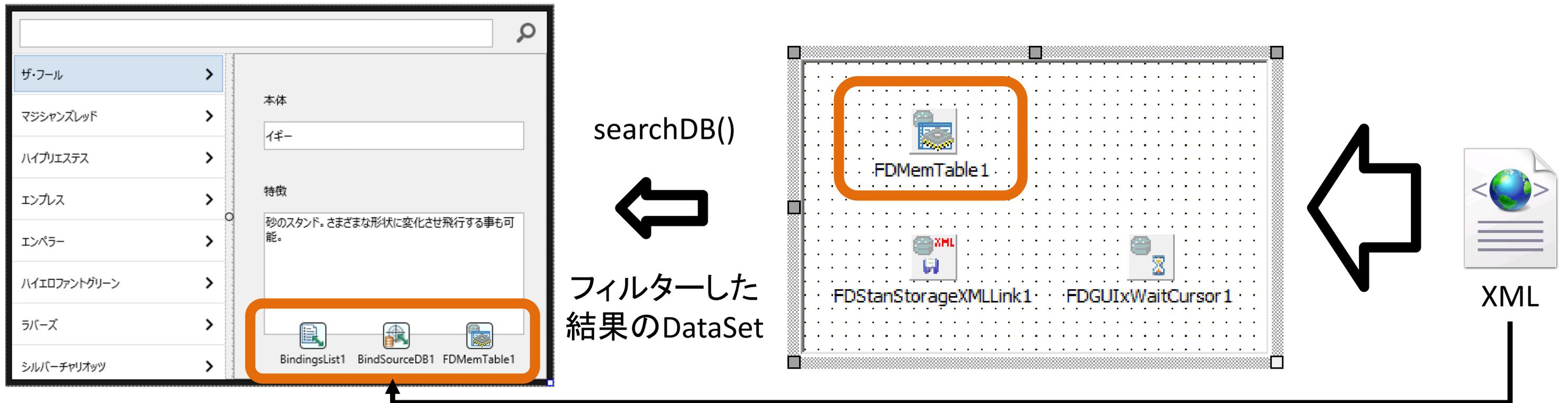
1: 23 クライアント ダイアレクト 3 トランザクションはアクティブです。 (読み取り) AutoDDL: ON

データ | プラン | 統計情報 | エラー

ST_ID	ST_NAME	RL_NAME	PART	INFO
300	ザ・フール	イギー	3	(MEMO)
301	マジシャンズレッド	モハメド・アヴドゥル	3	(MEMO)
302	ハイプリエステス	ミドラー	3	(MEMO)
303	エンプレス	ネーナ	3	(MEMO)
304	エンペラー	ホル・ホース	3	(MEMO)
305	ハイエロファントグリーン	花京院典明	3	(MEMO)
306	ラバーズ	綱入りのダン	3	(MEMO)
307	シルバーチャリオッツ	ジャン＝ピエール・ポルナレフ	3	(MEMO)
308	ストレンガス	フォーエバー	3	(MEMO)
309	ハーミットパープル	ジョセフ・ジョースター	3	(MEMO)
310	ホウィール・オブ・フォーチュン	ズイー・ズイー	3	(MEMO)
311	ジャスティス	エンヤ婆	3	(MEMO)

...¥Desktop¥DataModuleSampleFMX¥JODATA.IB

DMを初めて使う方へ(XML)



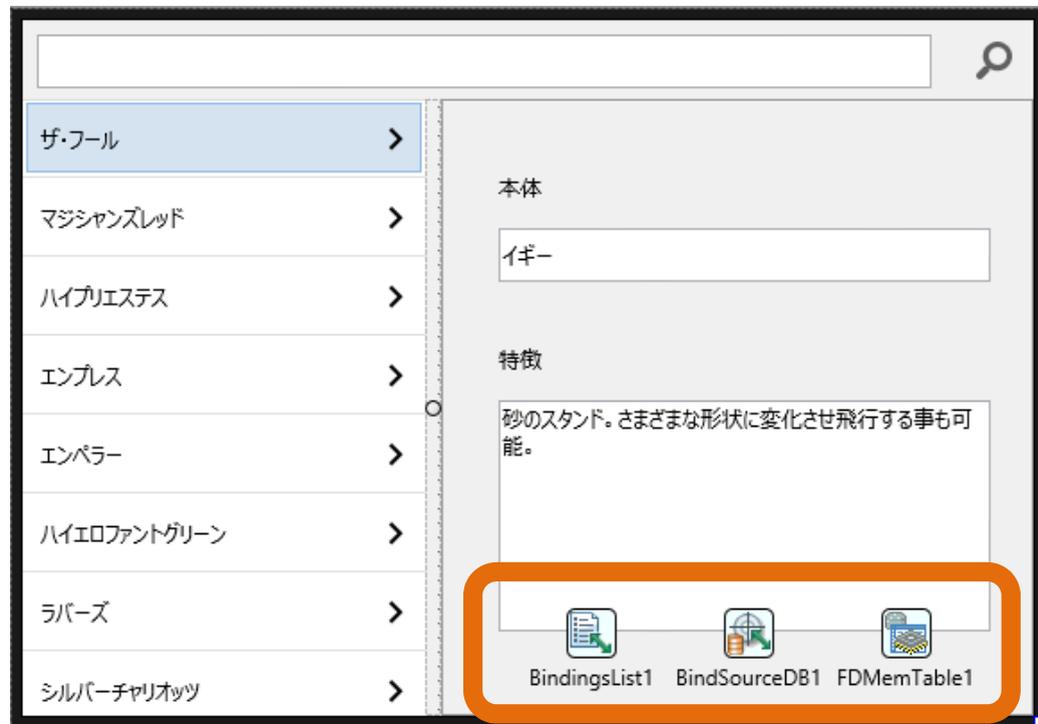
XMLファイルは**デザイン時**にも利用する

本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

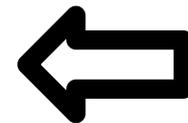
DMを初めて使う方へ(InterBase)



①デザイン時にはサンプルデータにXMLを用いつつ、

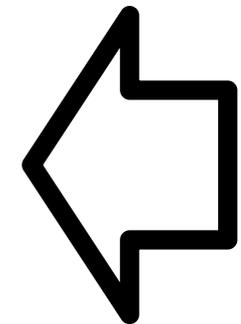
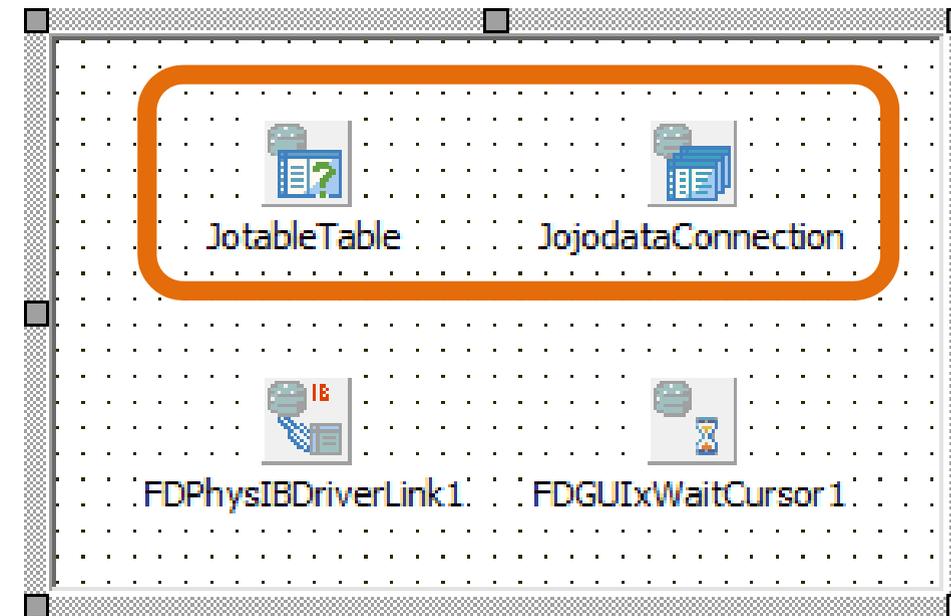


searchDB()



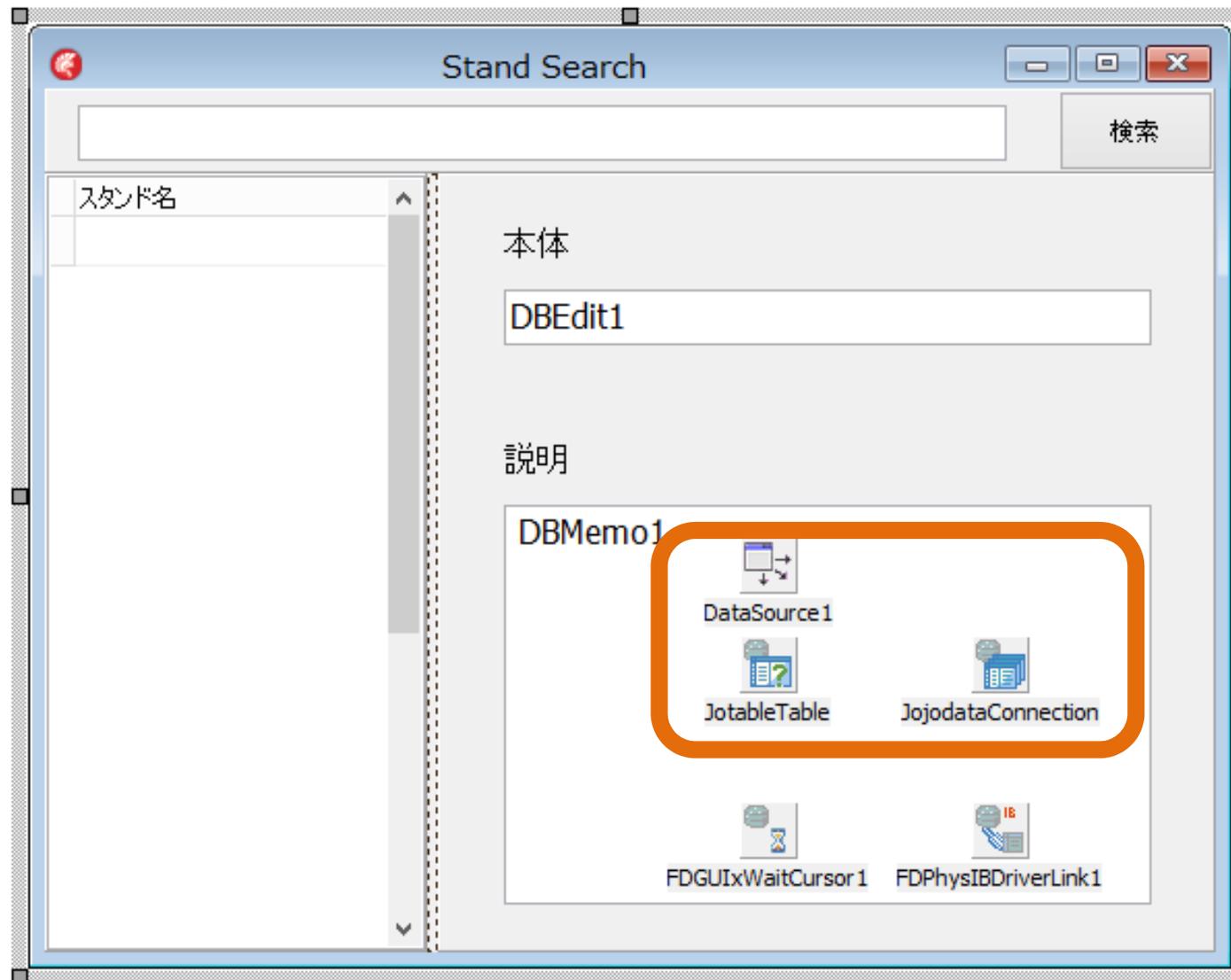
フィルターした
結果のDataSet

②実行時にはデータベースを用いるので開発が楽！

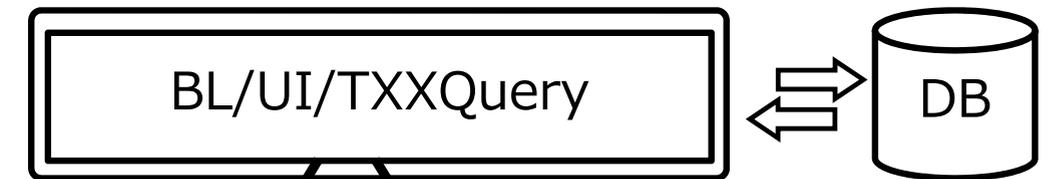


InterBase

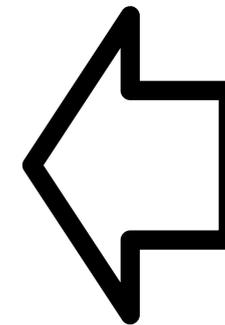
やっちまたなーの方へ



やっちまたなー状態

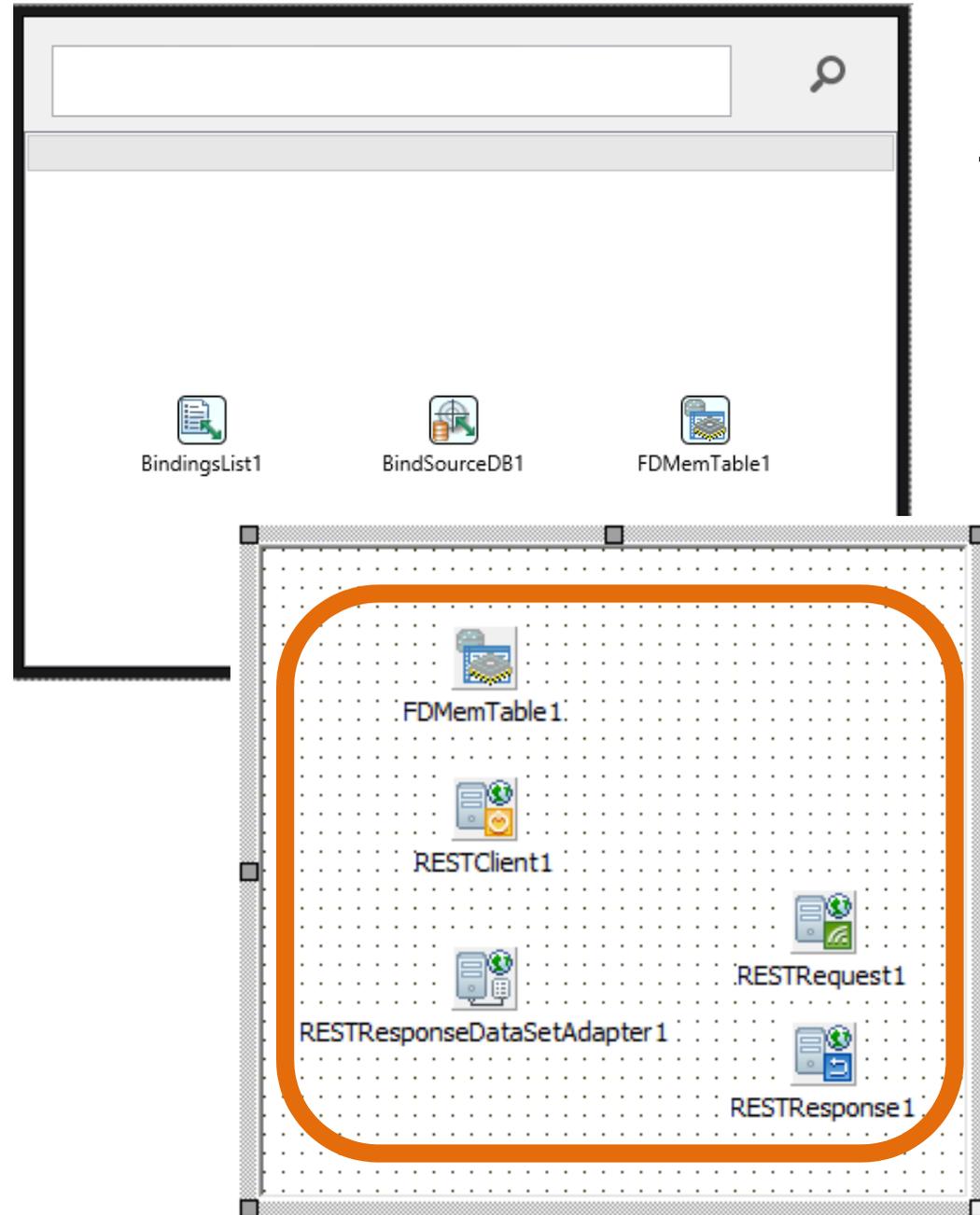


理想形



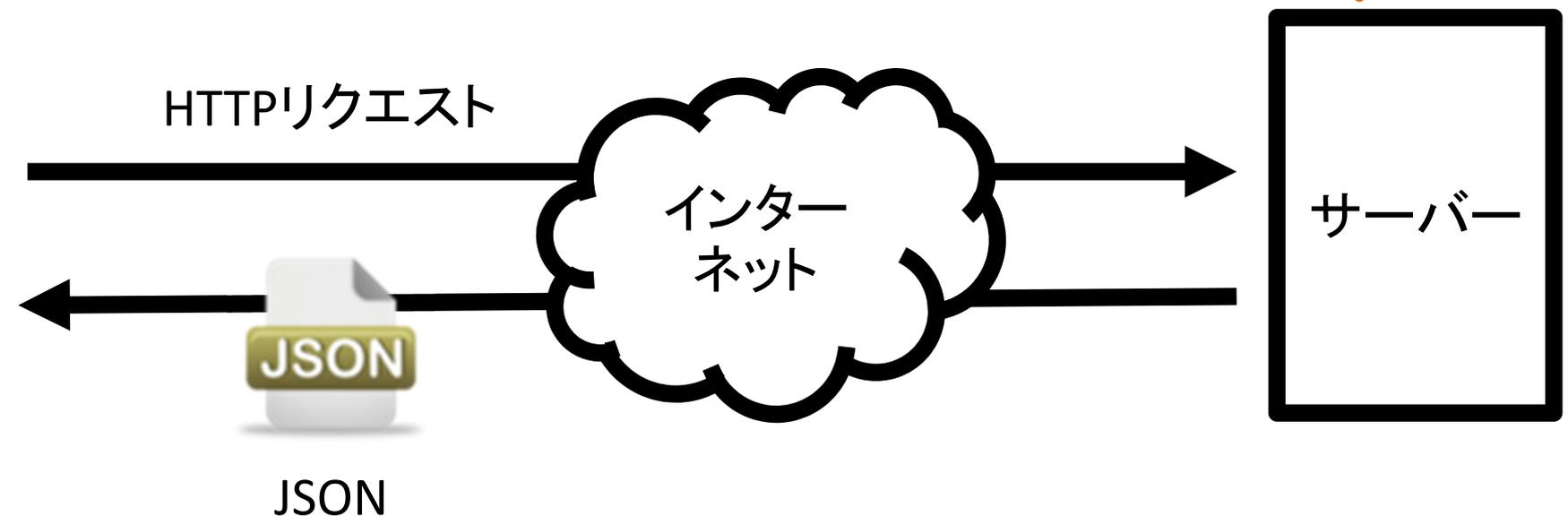
InterBase

その他:JSONデータのやりとり



ここでもsearchDB()を用いて
フォームとデータモジュールを繋いでいる！

今回のデモでは
以下のオープンデータサイトを利用しています
<http://www.songsterr.com/a/wa/api>

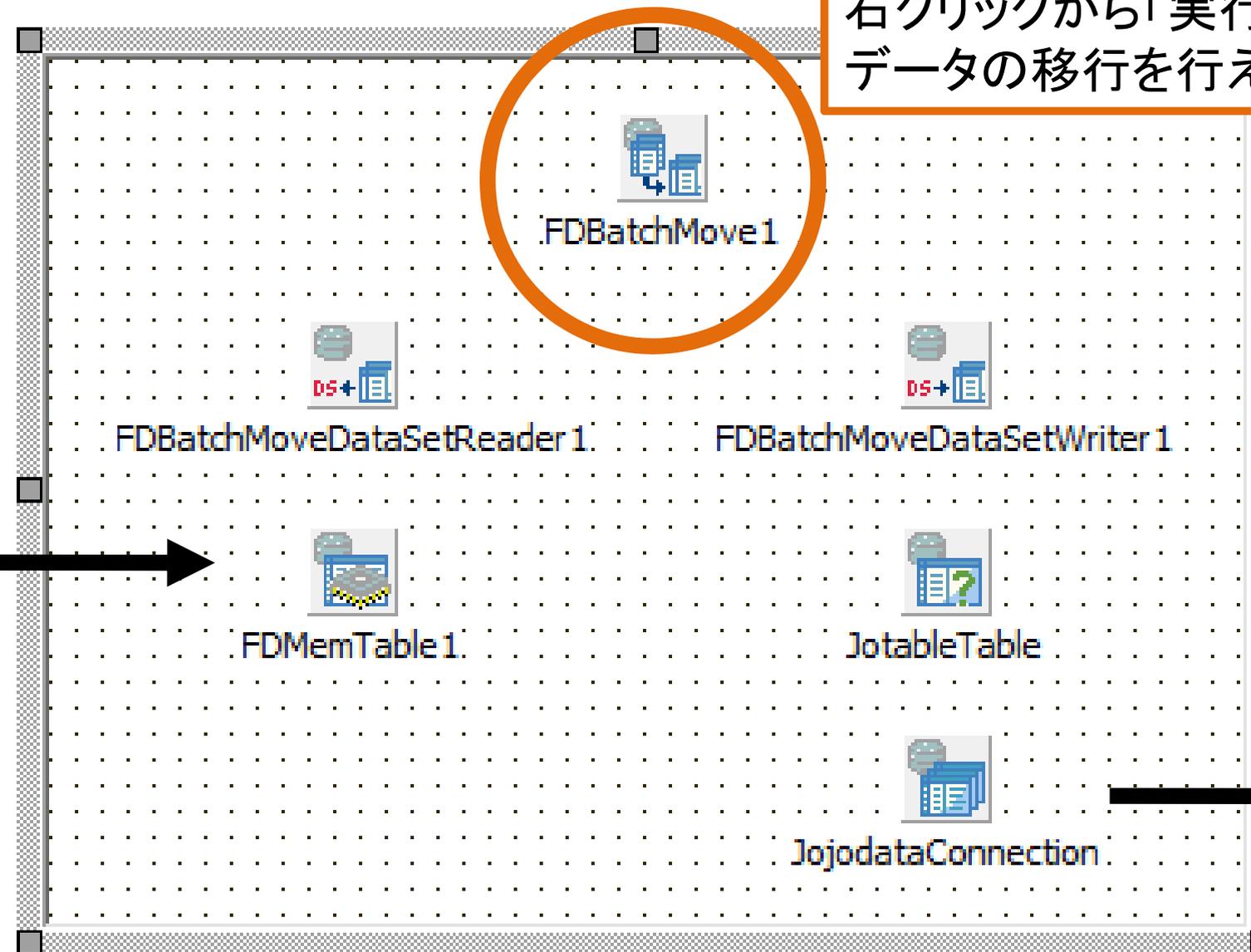


その他：種類の異なるデータソース間でデータ移行

コンポーネント1つで
異なるデータソース間で
データを移行できます。



XML



右クリックから「実行」で
データの移行を行えます



InterBase

移行プロジェクトでもモバイル開発でも必須！
今からでも間に合うデータモジュール活用法

まとめ

まとめ

- データモジュールは、非ビジュアルコンポーネントのコンテナで、たとえば、データアクセスコンポーネントを**隠蔽・集中管理**することができます。
→コレを使うと**UIとBLを切り分ける**ことができる。
- 「データモジュールが、ロジックとコードを切り分ける初めの一歩として再び重要」

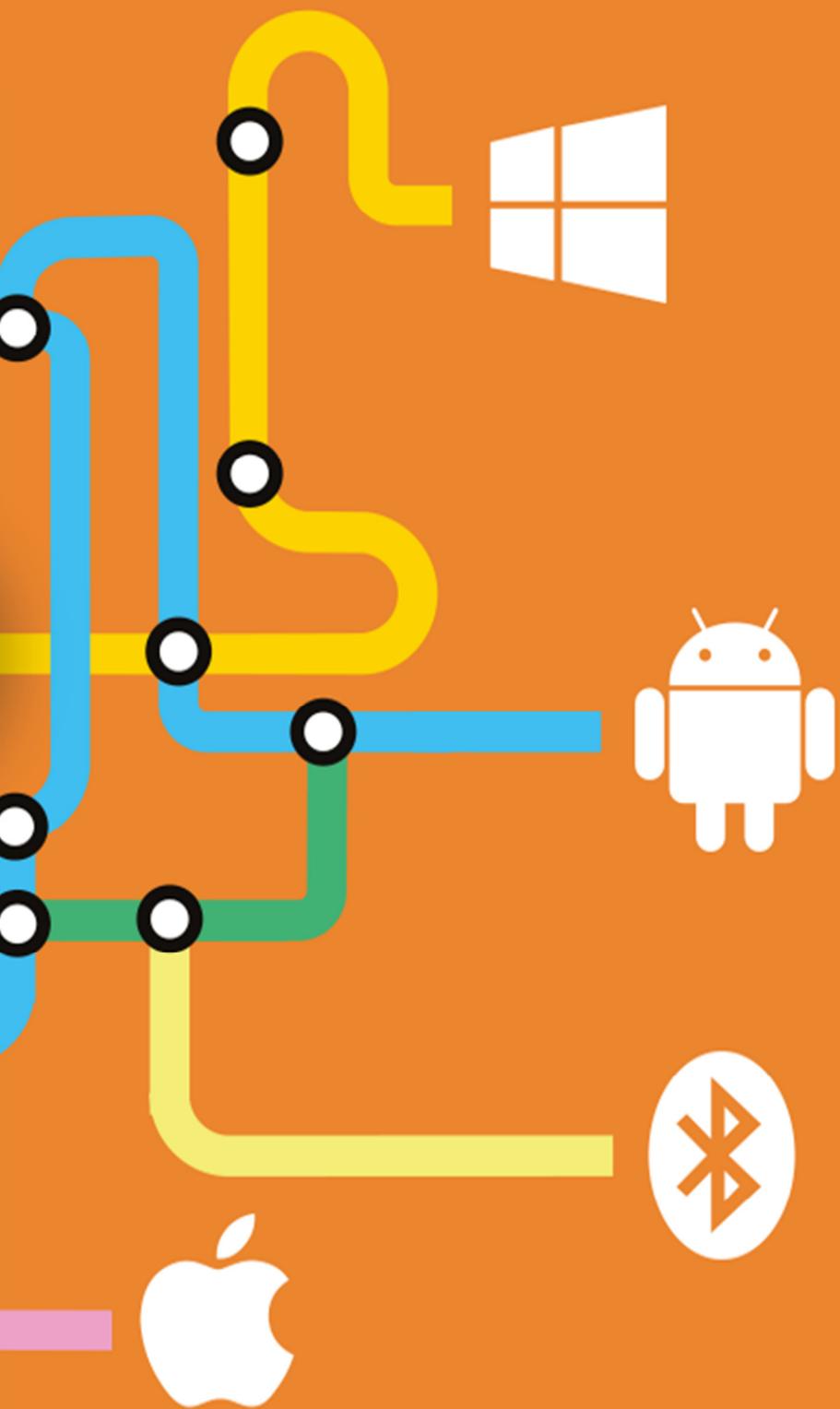
あとは、、、

- リファクタリングはこまめに行うこと
 - RADの本来の機能を十分に使いましょう
 - Delphi 2005以降、リファクタリングのための機能が数多くIDEに統合されています
- 分離は早めに行うこと
 - モック作成から分離していると、後々楽できる可能性が高い
例: 今回のデザイン時にXMLを使ったパターン
 - データモジュール内のprocedureの接続先を{IFDEF}で切り替えるなどが可能になる
- 複数のバージョンがテストできる環境を用意すること
 - 仮想環境構築のコストは極端に下がっています
 - 実機よりも仮想環境(E2, Azureなど含む)を使うことで、確実な検証ができます

ご静聴ありがとうございました

山本のメルマガ登録はこちらから！

http://services.dreamhive.co.jp/regist_mm



30TH EMBARCADERO DEVELOPER CAMP

第30回 エンバカデロ・デベロッパーキャンプ・ツアー

Thank you!