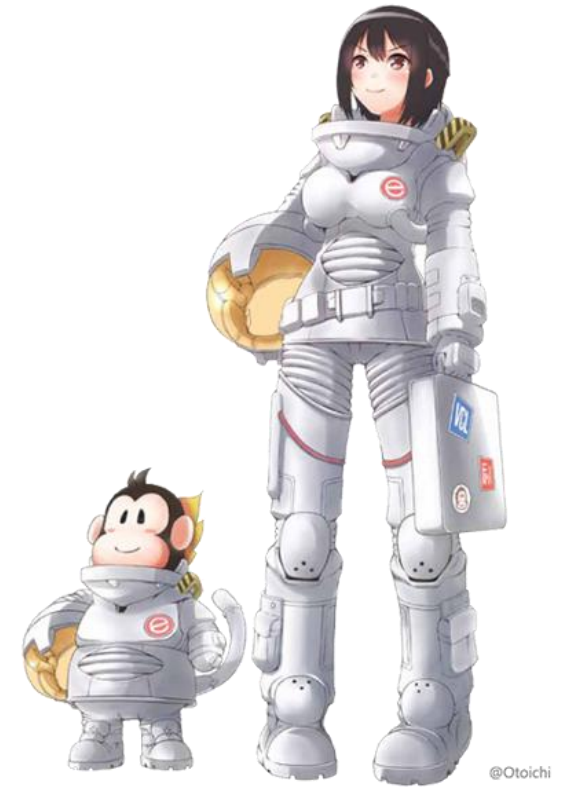


これが定番！RAD Studioで実践する既存 システムのモダナイズ

第34回 エンバカデロ・デベロッパーキャンプ

株式会社ドリームハイズ
代表取締役 山本 悟



embarcadero®
DEVELOPER CAMP

■はじめに

- 日々、新たな技術が生まれ、市場環境も変化し続けるIT業界。毎回新規開発ではキャッチアップも容易ではありません。そこで、既存システムをできる限り活用し、新規開発に比べてコスト削減・トラブル低減が見込める「モダナイゼーション」の手法が注目を集めています。今回、RAD Studioで既存アプリケーションをモダナイズする手法を、定番とも言えるいくつかのパターンを例に、デモを交えて紹介します。



■モダンイゼーション



embarcadero®
DEVELOPER CAMP

モダナイゼーションとは

- 「モダナイゼーション」とは、従来のシステムを有効活用しながら、最新のテクノロジーに適合できるように最新化（現代化）することです

モダナイゼーションで期待される効果



低コストで
最新技術に適合



全リプレイスによる
リスクを回避



短期間で求める
結果を実現

モダナイゼーションの一般的な手法

Rewrite

機能は変えずに書き直し。古いコードの移植にばかりこだわらずに、Delphiの生産性を活かして、書き直すことも手段のひとつ

Re-interface

古いアプリケーションは塩漬けにするものの、新しいシステムからアクセスできるようにインターフェイスを用意。中間サーバーはその解決策のひとつ

Re-host

稼働環境のみを移行。Delphiの場合、再コンパイルだけで新バージョンにという選択か？

Replace

古いシステム機能の一部を新しい技術で再構築。全体を作り直すのではないので、古いアーキテクチャで融通が利かない部分には有効な手段

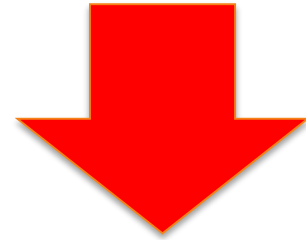
■ Re-host

- 稼働環境のみを移行。Delphiの場合、再コンパイルだけで新バージョンにという選択か？



Re-host

- 何も手を入れず 古いソースを新しい環境でビルドしたい
- BDEをそのまま使いたい
 - (DB構成はそのまま)
- 古いDelphiでビルドしたEXEを最新のWindows環境で動かしたい



問題の先り



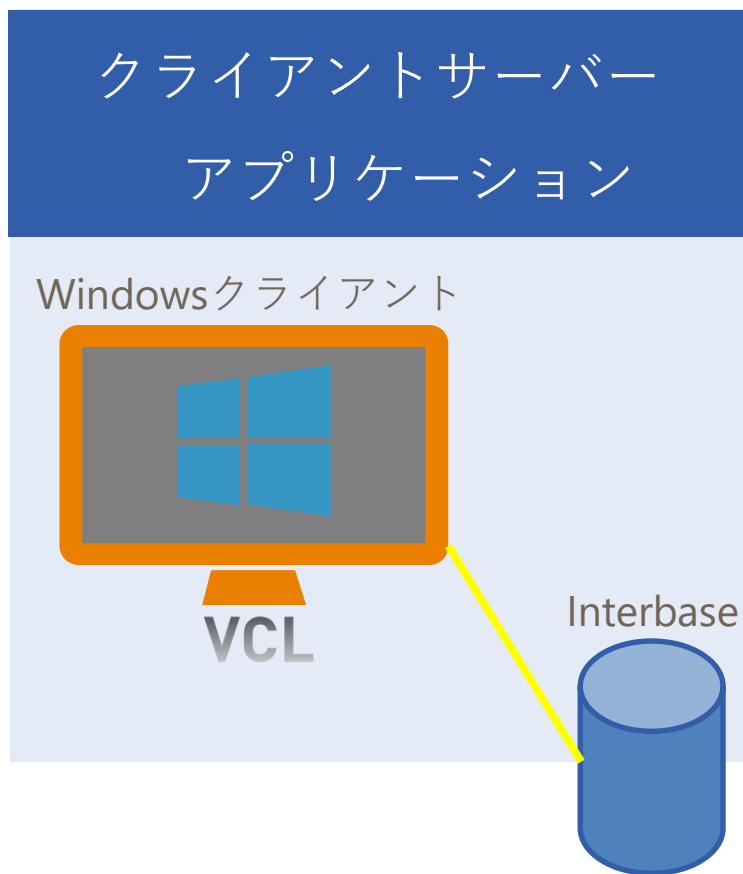
■ Rewrite

- 機能は変えずに書き直し。古いコードの移植にばかりこだわらずに、Delphiの生産性を活かして、書き直すことも手段のひとつ



既存のVCLアプリケーション

- 勤怠管理 VCL Windows C/Sアプリケーションです。



勤怠管理

ファイル(F) 管理(K)

1001 中原 中也

出勤 退勤

NUM	打刻日時	ステータス	部署	姓	名
10	2017/07/05 16:53:56	退勤	総務	国木田	独歩
9	2017/07/05 16:51:39	出勤	総務	国木田	独歩
8	2017/07/05 16:21:20	退勤	営業部	福沢	諭吉
7	2017/07/05 16:21:18	出勤	営業部	福沢	諭吉
6	2017/07/05 16:21:12	出勤	開発部	中原	中也
5	2017/07/05 16:21:08	退勤	開発部	中原	中也
4	2017/07/05 16:19:32	出勤	営業部	福沢	諭吉
3	2017/07/05 16:05:43	出勤	開発部	中原	中也
2	2017/03/03 3:03:30	出勤	開発部	中原	中也
1	2017/03/03 3:03:30	出勤	開発部	中原	中也

ソースコードが無いケースもあります。
当時のコンポーネントが無い

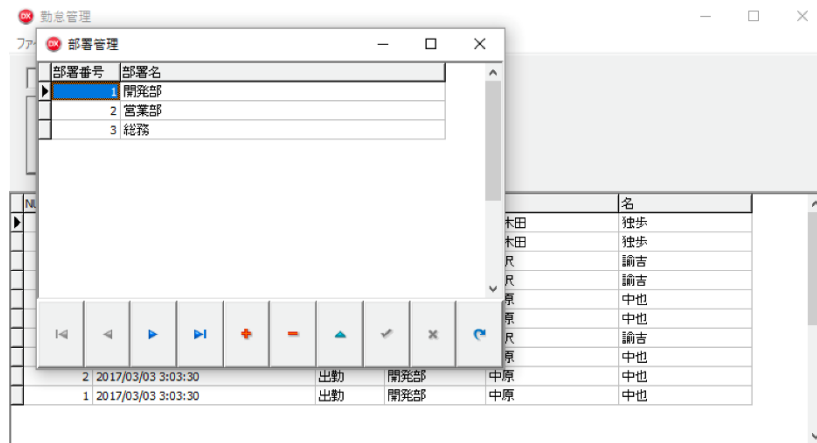
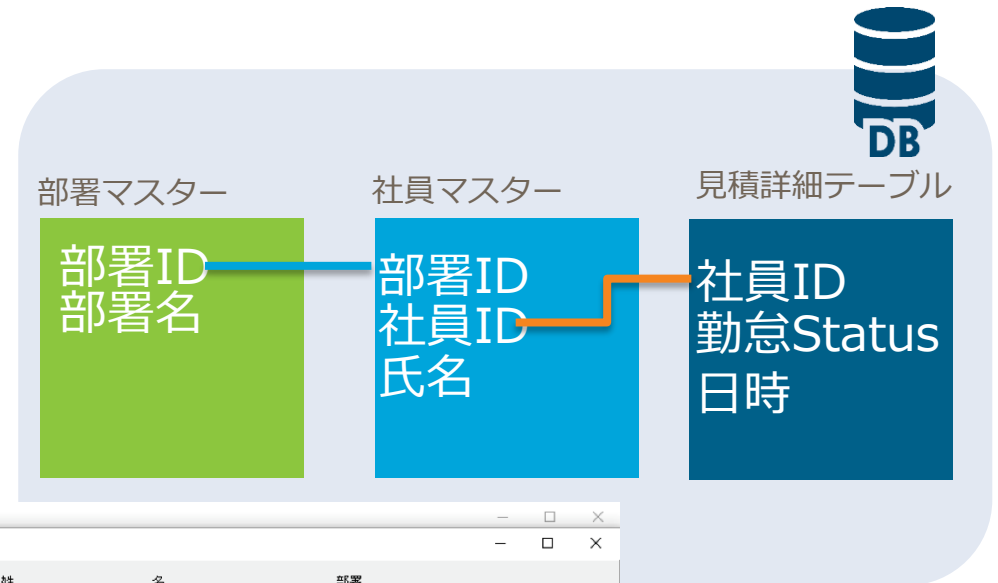
勤怠管理アプリケーション(VCL)

■ 勤怠システムの画面 (Delphiで構築したWindowsアプリケーション)

- ・ 勤怠データ(メイン)
- ・ 社員管理マスター
- ・ 部署管理マスター

■ 使用機能等

- ・ データベース: InterBase



■ Re-interface

- 古いアプリケーションは塩漬けにするものの、新しいシステムからアクセスできるようにインターフェイスを用意。中間サーバーはその解決策のひとつ



■ さまざまなDB接続モダンイゼーション

- 既存のVCL少しの変更でさまざまなDBへのモダンイゼーション



さまざまなDBに接続の為のモダナイゼーション

- 既存VCLのアプリケーションから更にさまざまなDB接続をしていく

勤怠管理

ファイル(F) 管理(K)

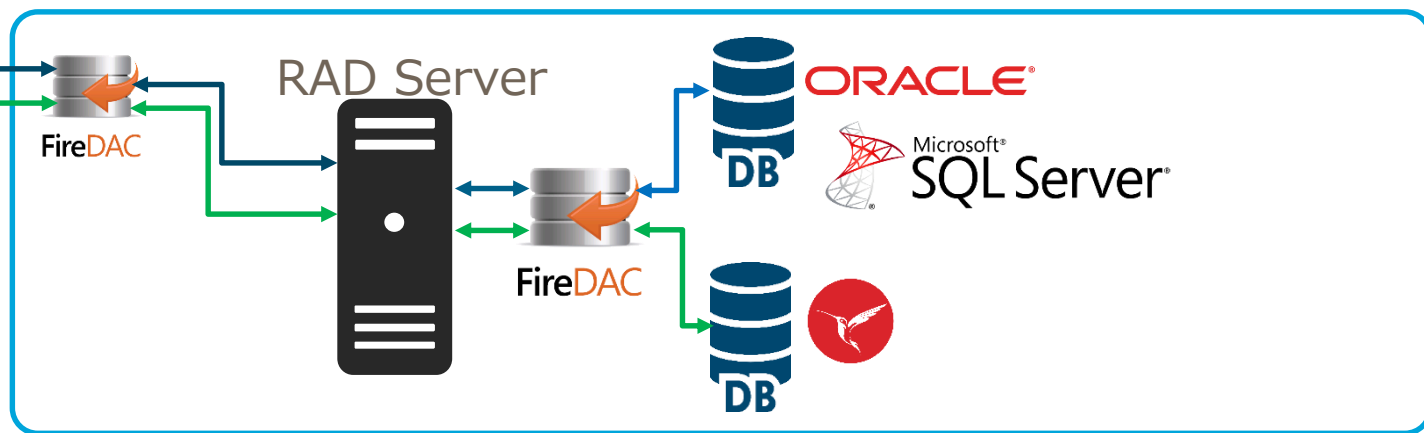
出勤 退勤

LOG_NUM	KINTAI_DATE	ステータス	部署	姓	名
4	2017/07/18 15:45:24	退勤	開発部	相縣	和食
3	2017/07/18 15:45:17	退勤	営業部	井之上	和弘
2	2017/07/18 15:45:15	出勤	営業部	井之上	和弘
1	2017/07/18 15:45:10	出勤	開発部	相縣	和食

接続をちょい足し

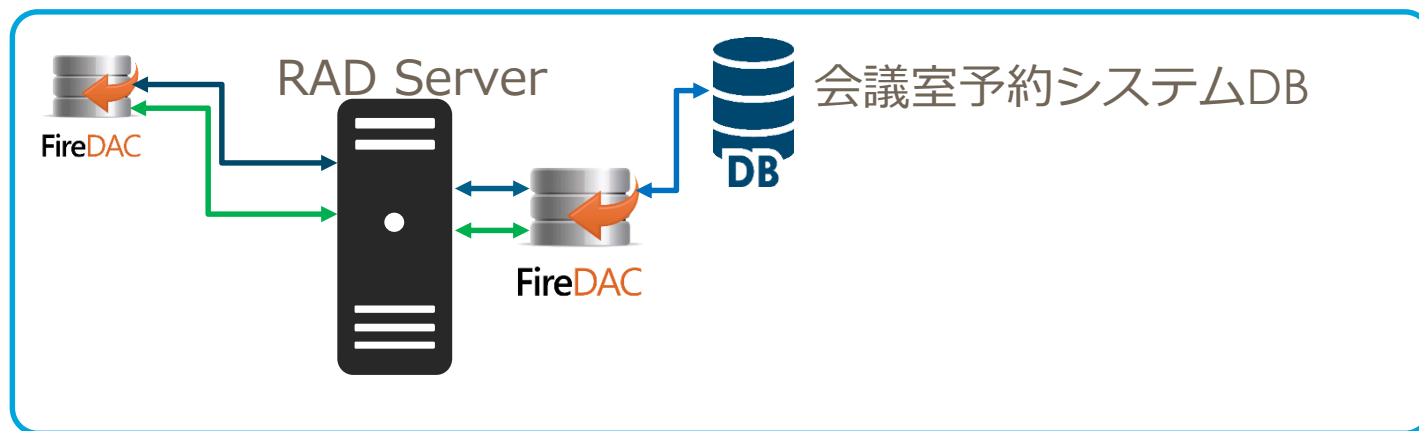
DB
既存DB

FireDAC



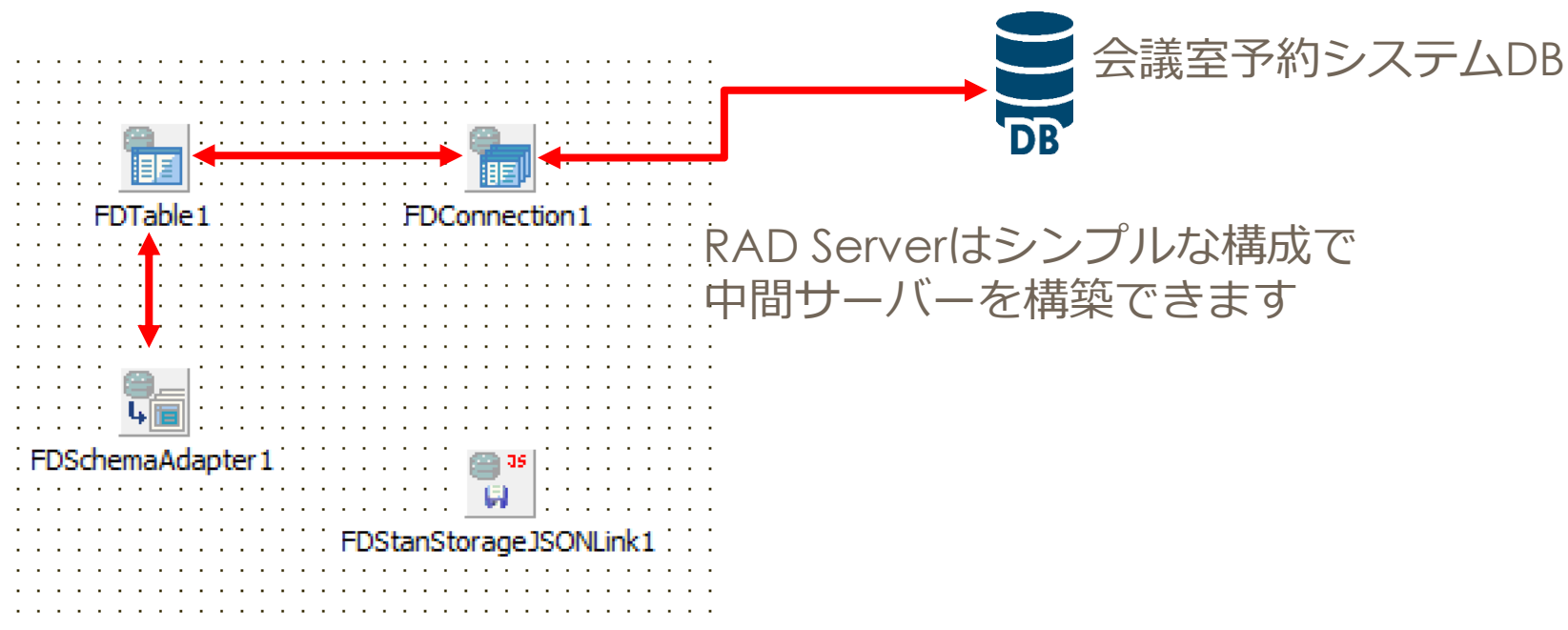
勤怠管理システムに外部のDB接続

- 既存の勤怠管理システムに会議室管理システムDB接続
 - 直接接続するのはセキュリティに問題がある
- 中間サーバーを立てる事により勤怠管理システムと分離



中間サーバーを立てる事により勤怠管理システムと分離

- RAD Server側のシステム構築



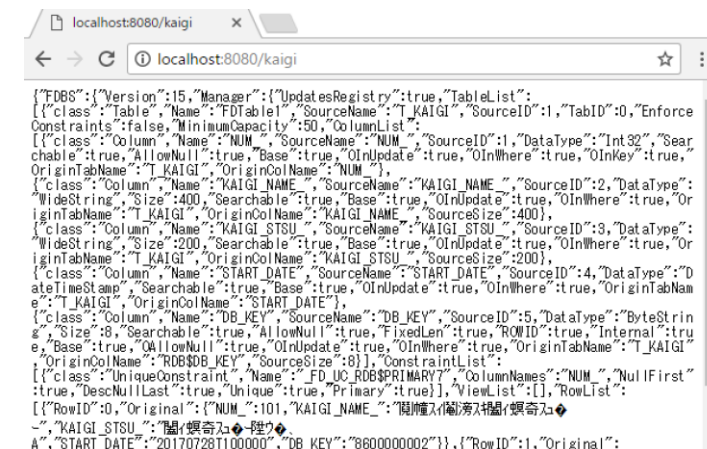
中間サーバーを立てる事により勤怠管理システムと分離

■ コード

```

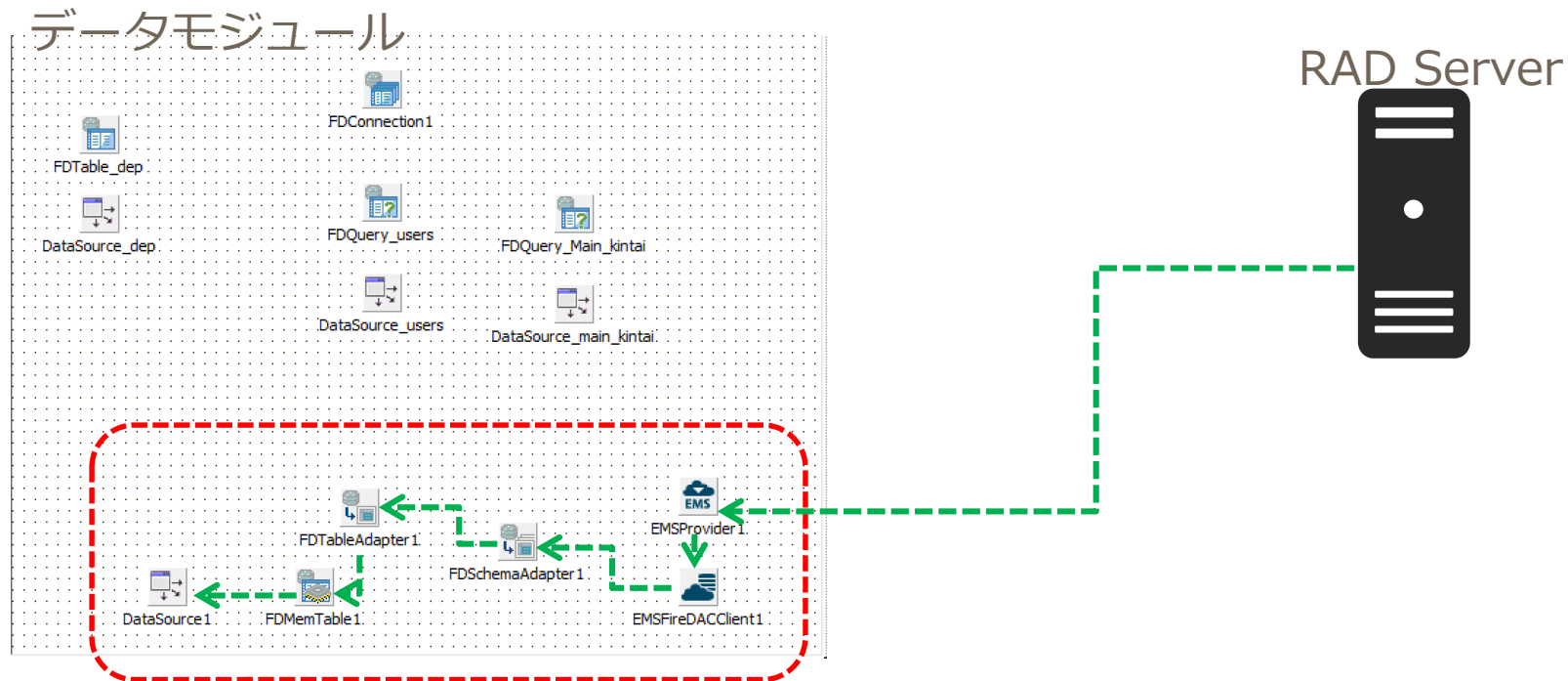
procedure TKaigiResource1.Get(const AContext: TEndpointContext; const ARequest: TEndpointRequest; const
AResponse: TEndpointResponse);
var
    mm: TMemoryStream;
begin
    // Sample code
    mm := TMemoryStream.Create;
    FDConnection1.Connected := True;
    FDTable1.Active := True;
    FDSchemaAdapter1.SaveToStream(mm, TFDStorageFormat.sfJSON);
    mm.Position := 0;
    AResponse.Body.SetStream(mm, 'application/json', True);
end;

```



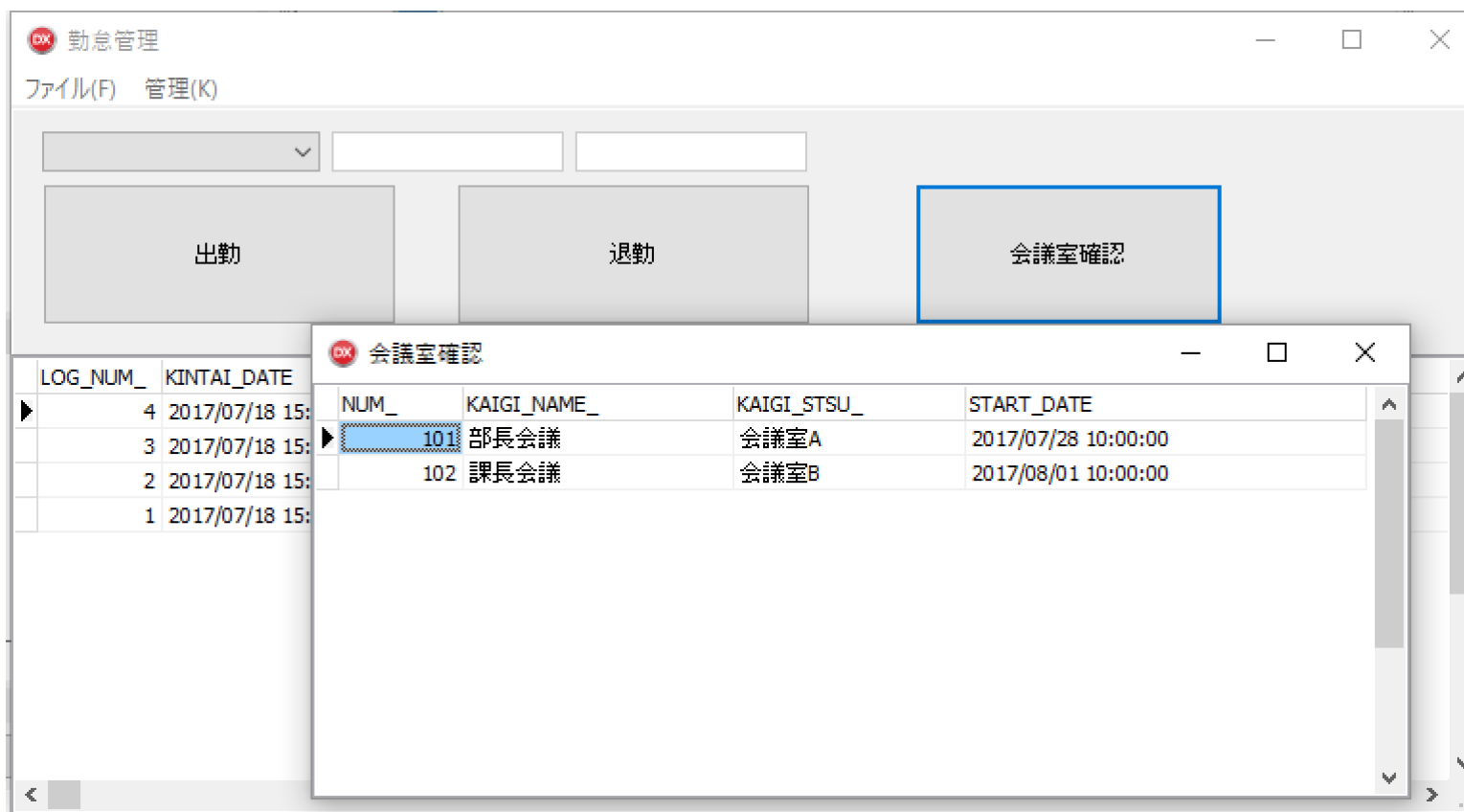
データモジュールにTEMSFireDACClientの追加

- クライアント側データモジュールにコンポーネントを追加
 - TEMSProvider, TEMSFireDACClient, TFDSchemaAdapter, TFDDTableAdapter, TFDMemTable, TDataSource



勤怠管理システムに会議室情報を表示

- 勤怠管理システムに会議室画面を追加



印刷方法のモダナイゼーション

- 既存のVCL少しの変更でさまざまな帳票を出力するモダナイゼーション



印刷方法のモダナイズ

- 既存のVCLアプリにPDF出力機能を追加
 - スマートフォンで印刷するの？



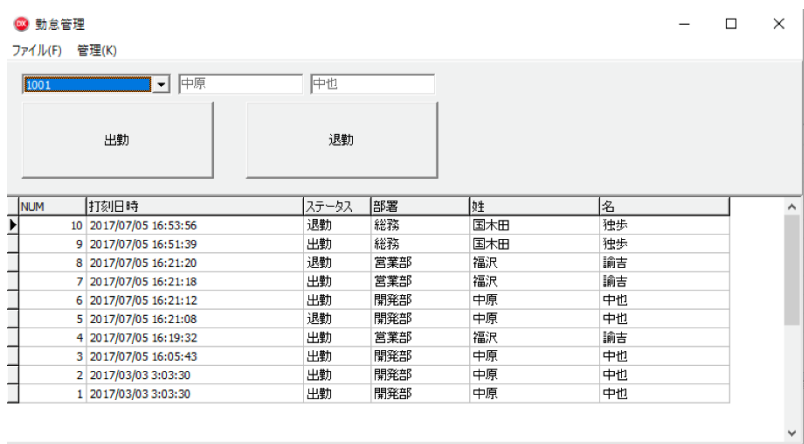
■ Replace

- 古いシステム機能の一部を新しい技術で再構築。全体を作り直すのではないので、古いアーキテクチャで融通が利かない部分には有効な手段

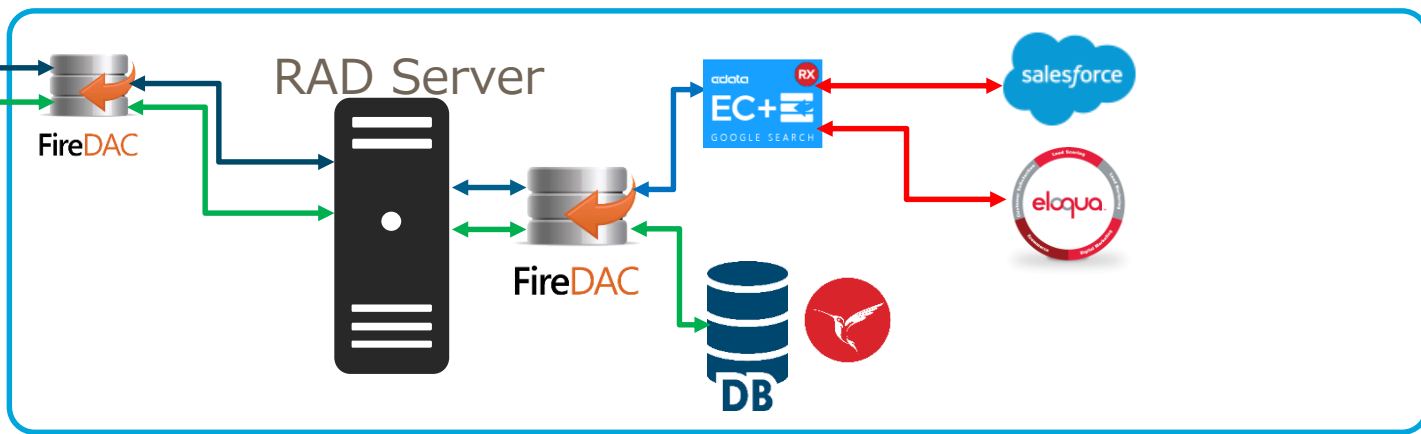
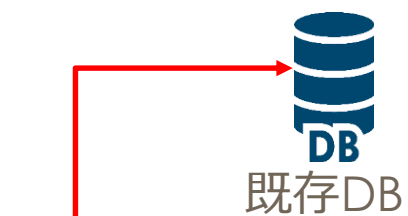


クラウド接続でモダナイゼーション

- 既存VCLのアプリケーションから
 - RAD Server + Enterprise Connect
 - クラウドDB接続をしていく



RAD Serverを利用し
Enterprise Connector側のデータを
間接的にデータ取得する事で
セキュリティ担保



まとめ

- Rewrite
 - 既存VCLアプリ DelphiでRewrite
- Re-interface
 - RAD Serverを用いた 既存VCLでさまざまなDBと接続
 - Fast Reportを使いモダンな印刷方法
- Replace
 - (仮) FireDAC + Enterprise Connectorを使い
既存VCLでクラウドデータを処理

THANKS!

www.embarcadero.com/jp

第34回 エンバカデロ・デベロッパーキャンプ