

【セッションNo. 3】

スマートデバイスの運用を考慮した 「IBM i 接続制御テクニック」

株式会社ミガロ.
システム事業部 プロジェクト推進室
小杉 智昭

【アジェンダ】

1.はじめに

2.スマートデバイスからのIBM i 接続

- 2-1.スマートデバイスのネットワーク環境課題点
- 2-2.スマートデバイスで使用されるアプリケーション
- 2-3.ネットワーク切断後の動作

3.ネットワーク切断を考慮したIBM i の接続制御

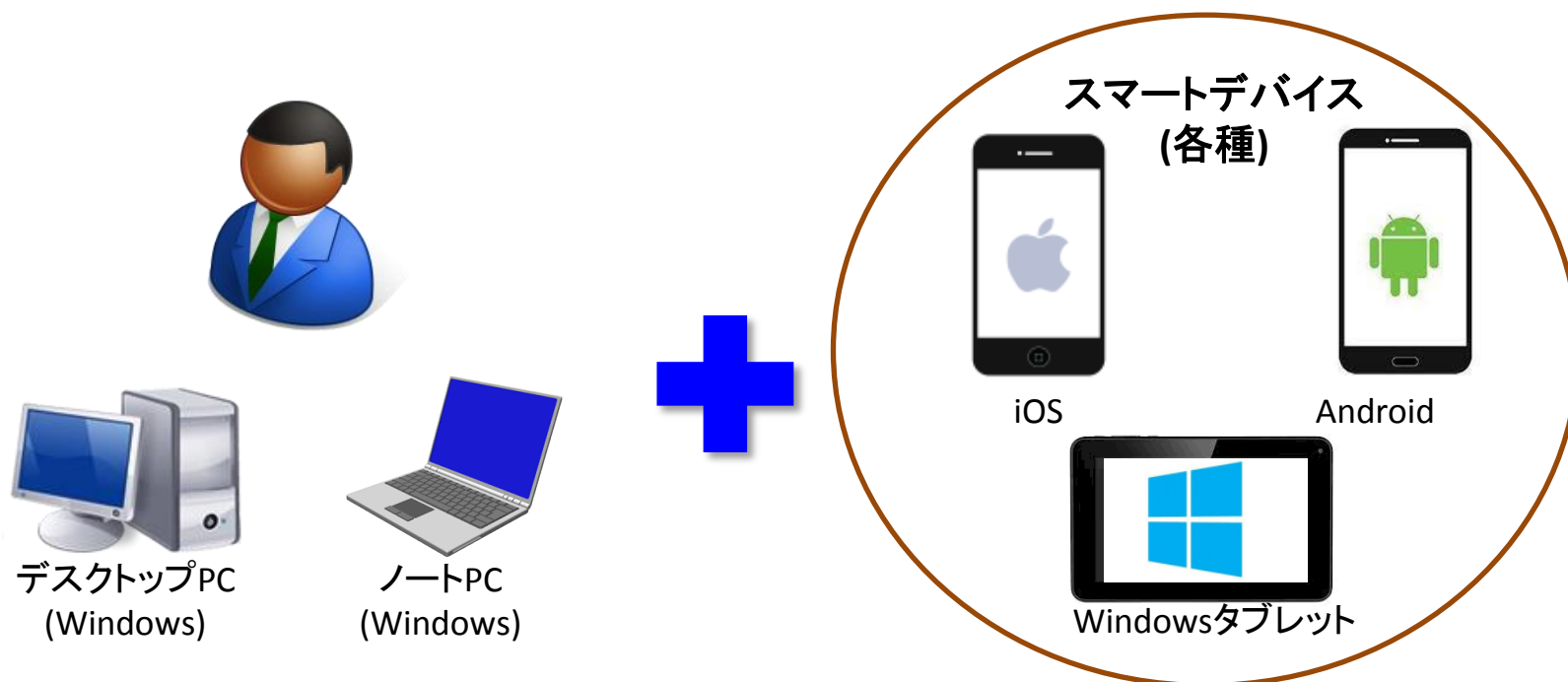
- 3-1.アプリケーションでIBM i 接続を制御する手法
- 3-2.ポイント① IBM i 接続チェック
- 3-3.ポイント② IBM i 接続制御
- 3-4.Webアプリケーションにおける再接続の工夫(補足)

4.まとめ

1.はじめに

- スマートデバイスの企業利用状況

これまでシステム運用するデバイスはWindowsのPCが主な対象でしたが、ここ数年でiOSやAndroid、Windowsを搭載したスマートデバイスの導入企業が急激に増加

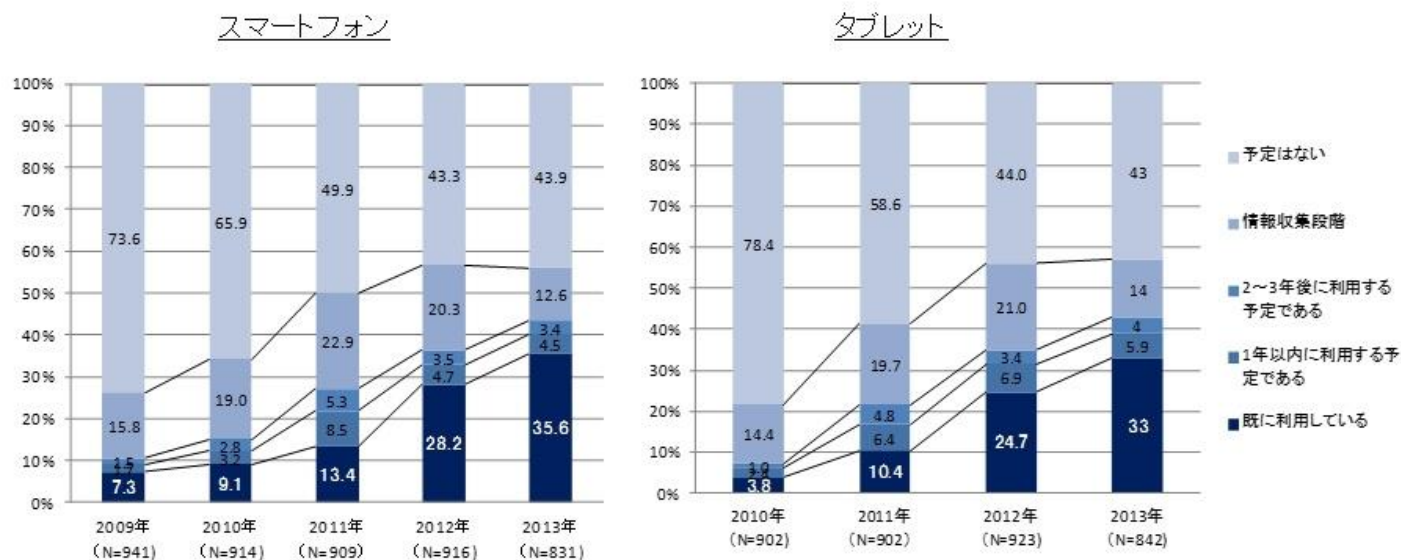


1.はじめに

・スマートデバイスの企業利用状況

企業アンケートではスマートデバイスを「既に利用している」は3割、「予定を含める」と5割を上回る

Q) 貴社の従業員は現在、業務で会社が支給するスマートフォン、タブレット端末を利用していますか。もしくは今後、利用する予定はありますか。(それぞれひとつだけ)



出所) NRI「企業情報システムとITキーワードに関する調査」
 ※「わからない」「無回答」選択は有効回答から除外した

「2014年、スマートデバイスの企業活用はセカンドステージへ IT Leaders」より抜粋

<http://it.impressbm.co.jp/articles/-/11183>

1.はじめに

• デバイスの違いによる利用シーン

スマートデバイスは、これまでのデスクトップ・ノートPCと機器が違うだけでなく、その特徴から利用されるシーンや場所・環境が大きく異なる(広がる)

デスクトップ・ノートPC



スマートフォン・タブレット



① 端末を利用する環境

【固定環境】

- ・事務所にある自分の席
- ・工場などに設置された特定の場所
- ・店舗の事務・会計

【携帯環境】

- ・事務所内どこでも、外出先
- ・工場や倉庫内のどこでも
- ・店舗内のどこでも

これまでのシステムの
利用が届かなかった
場面で活躍

1.はじめに

• デバイスの違いによる利用シーン

デスクトップ・ノートPC

スマートフォン・タブレット

②利用するときの状態

座って(止まって)PCだけを操作

移動や他作業をしながら操作

③ネットワーク接続

【安定】

接続:有線、無線



固定された場所で使用する為、社内のネットワークが安定した環境で、システムを利用

ネットワークが切断されることは基本的に考慮が不要

【不安定】

接続:無線



移動して使用する為、場所によってはネットワークが不安定な環境でシステムを利用

ネットワークが切断される(IBMiとの接続が切れる)場面を考慮する必要がある

2.スマートデバイスからのIBM i 接続

2.スマートデバイスからのIBM i 接続

2-1.スマートデバイスのネットワーク環境課題点

ネットワークが切断されてしまうと、IBM i への接続も切断される。
スマートデバイスでは、どういった場合にネットワークが切断されてしまうのか？

携帯・移動しての利用が特徴である反面
いつも安定したネットワークが利用できるとは限らない



屋内であっても、無線が届かない場所や
あるいは周りの使用状況で通信が弱くなることもある

基本的に電源がつながっておらず、バッテリーの省電力化が
重視されるため、スリープ状態への移行が早い



WindowsのOSでは、スリープ状態になると
ネットワークの機能を停止して切断されてしまう

2.スマートデバイスのIBM i 接続

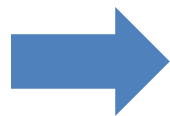


2-1.スマートデバイスのネットワーク環境課題点

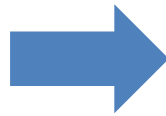
ネットワークが切断されるとどうなってしまうか？

IBM i への接続が切断されて IBM i 上のジョブが終了してしまう為、
ネットワークに再接続してから操作を行っても 正しく処理が行えない場合がある

ネットワーク
接続中



ネットワーク
切断時



ネットワーク
再接続後



ネットワークがつながったのに
正しく処理が行えない

2.スマートデバイスからのIBM i 接続

2-2.スマートデバイスで使用されるアプリケーション

IBM i への接続はネットワークに依存しますが、アプリケーションの種類によって、対象のデバイスや接続構成が異なる

モバイルネイティブアプリケーション

- iOS、Androidに直接インストールして、デバイス機能と連携動作できるアプリケーション。IBM i とはDataSnapサーバを経由して接続する。

Client/Serverアプリケーション

- Windowsに配置して動作するEXE形式のアプリケーション。(従来のWindowsアプリケーション) IBM i とは直接接続する。

Delphi/400は全てのアプリケーション形式をサポート

Webアプリケーション

- デバイスには依存せず、Webブラウザを使いURLにアクセスして動作するアプリケーション。IBM i とはWebサーバを経由して接続する。

接続制御を行うためには、デバイス上にアプリケーションが必要

2.スマートデバイスからのIBM i 接続

2-2.スマートデバイスで使用されるアプリケーション

＜モバイルネイティブアプリケーションのIBM i 接続構成＞

スマートデバイス上のアプリケーションから、DataSnapサーバを経由してIBM iへ接続し、処理を行う。

スマートデバイス



モバイルネイティブ
アプリケーション

デバイス上に
アプリがある
||
接続制御可能

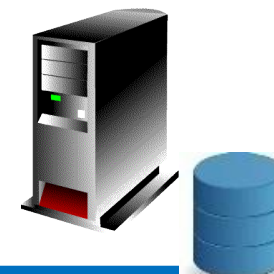


DataSnapサーバ



DataSnap
アプリケーション

IBM i



無線ネットワーク環境に依存する接続は
DataSnapサーバへの接続

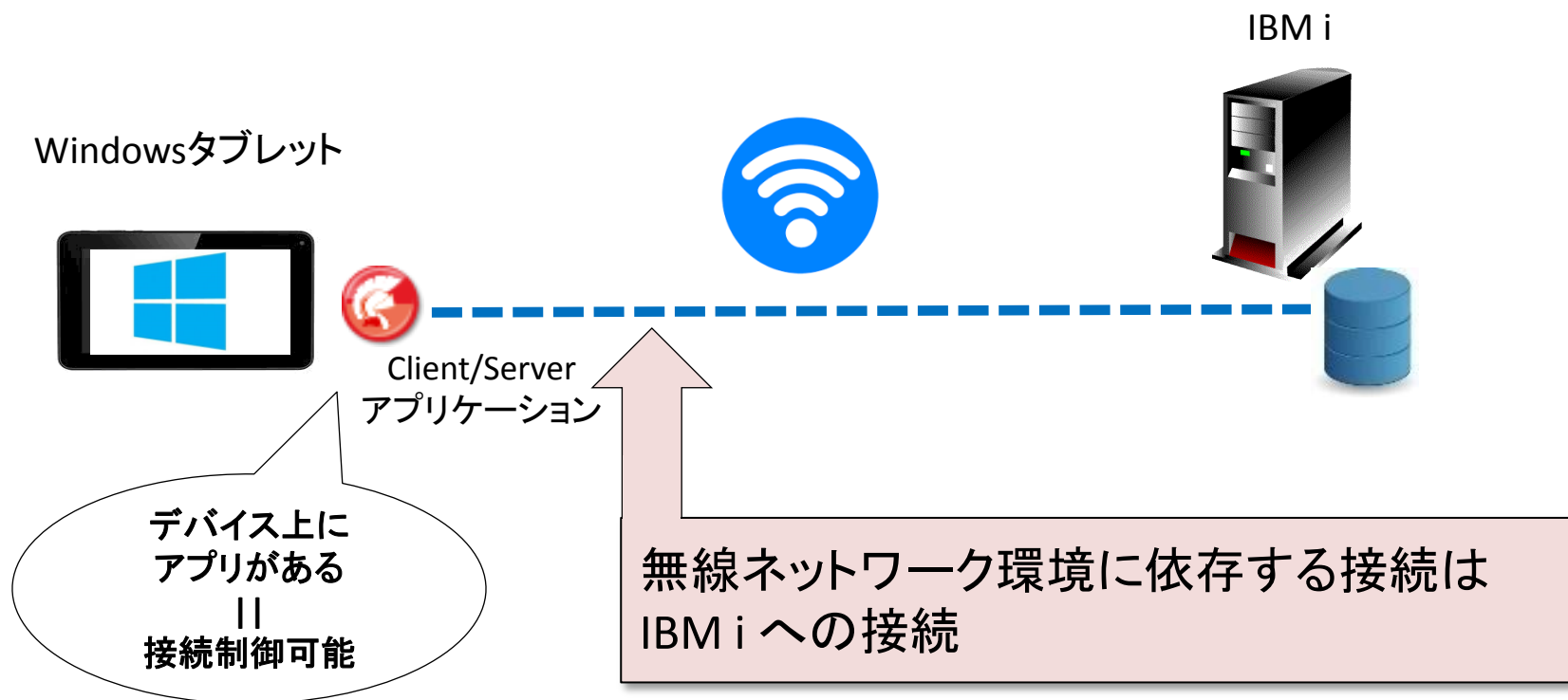
2.スマートデバイスからのIBM i 接続

2-2.スマートデバイスで使用されるアプリケーション

＜Client/ServerアプリケーションのIBM i 接続構成＞

Windowsタブレット上のアプリケーションから、

直接IBM iへ接続して処理を行う。(PCがタブレット端末に変わっただけの違い)

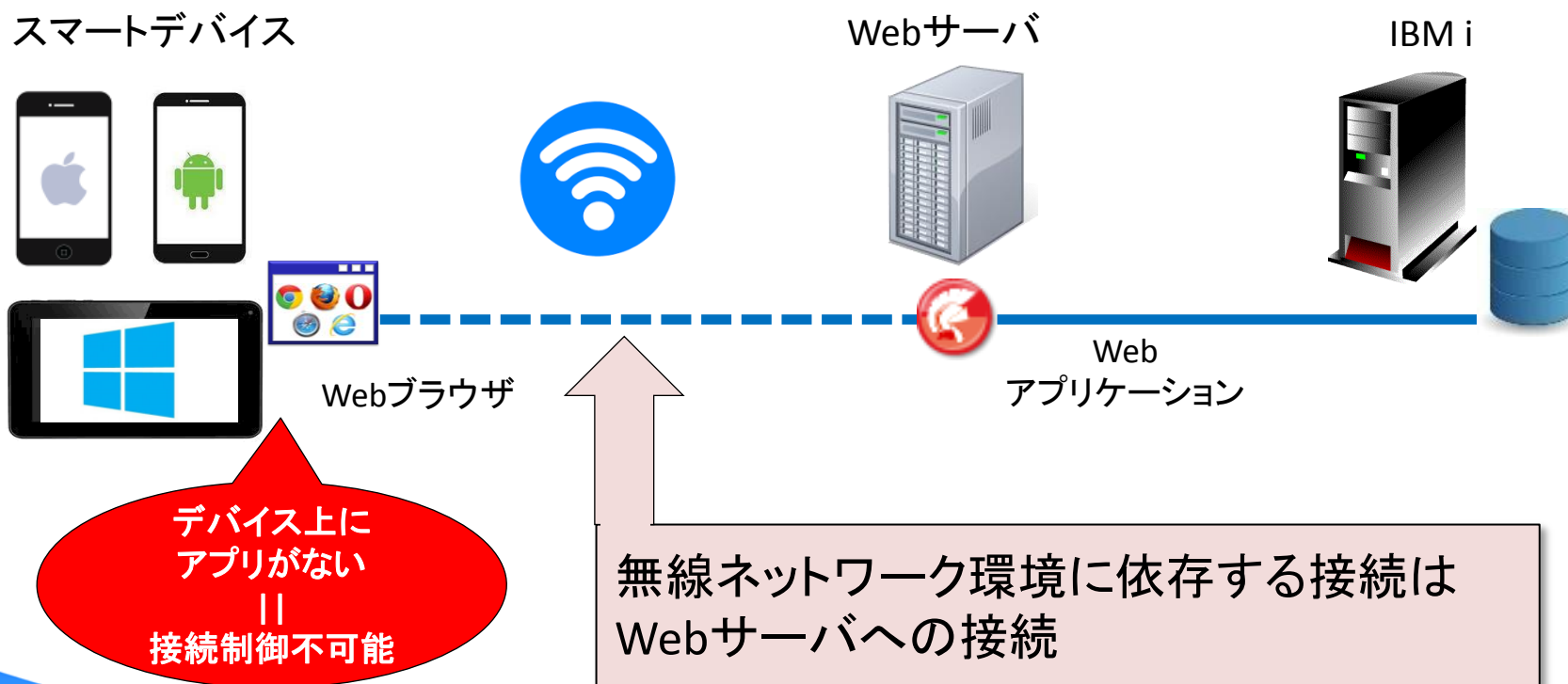


2.スマートデバイスからのIBM i 接続

2-2.スマートデバイスで使用されるアプリケーション

＜WebアプリケーションのIBM i 接続構成＞

スマートデバイス上のWebブラウザから、Webサーバ上のアプリケーションを経由してIBM iへ接続し、処理を行う。



2.スマートデバイスからのIBM i 接続

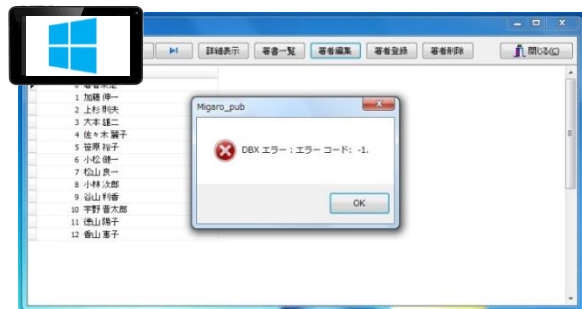
2-3.ネットワーク切断後の動作

IBM i へ接続するアプリケーションは一度切断されると処理がエラーとなる

モバイルネイティブアプリケーション



Client/Serverアプリケーション



Webアプリケーション



2.スマートデバイスからのIBM i 接続

2-3.ネットワーク切断後の動作

IBM i へ接続するアプリケーションは一度切断されると処理がエラーとなる

モバイルネイティブ アプリケーション

- デバイス側にアプリケーションがある
→ アプリケーション側で接続制御が可能

この制御を中心に
テクニックをご説明

Client/Server アプリケーション

- デバイス側にアプリケーションがある
→ アプリケーション側で接続制御が可能

Web アプリケーション

- デバイス側にアプリケーションがない
→ 再ログインが必要
接続構成で工夫(後述)

3. ネットワーク切断を考慮したIBM i の接続制御

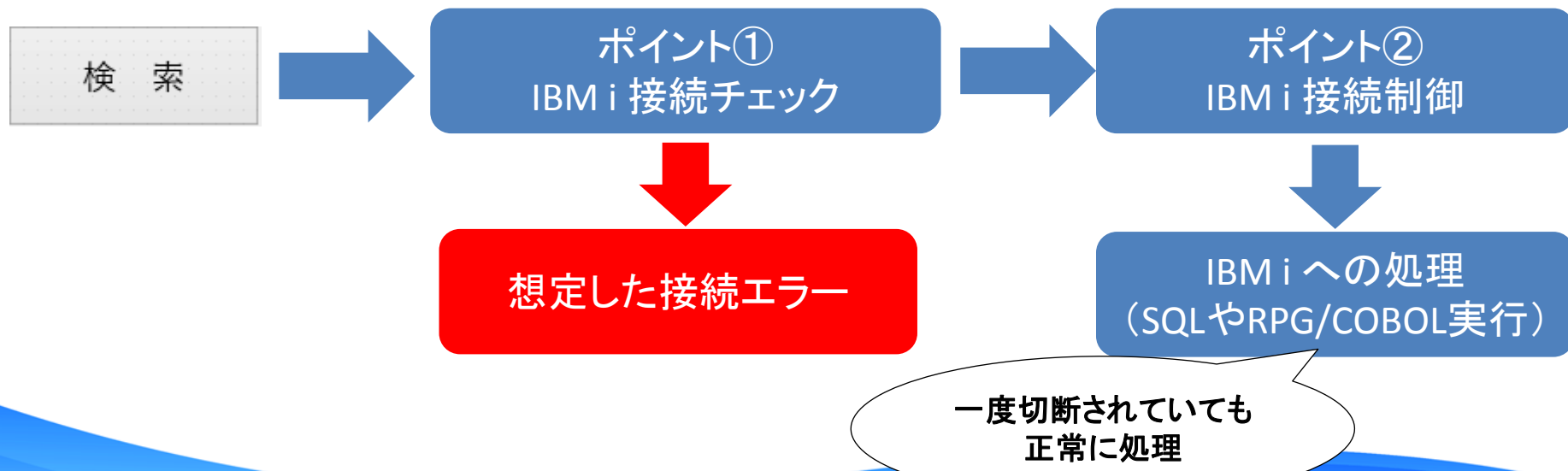
3. ネットワーク切断を考慮したIBM i の接続制御

3-1. アプリケーションでIBM i 接続を制御する手法

IBM i との接続制御を考慮しないプログラム例



IBM i との接続制御を考慮したプログラム例



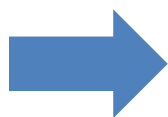
3. ネットワーク切断を考慮したIBM i の接続制御



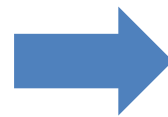
3-1. アプリケーションでIBM i 接続を制御する手法

IBM i との接続制御を考慮したプログラム実行例

ネットワーク
接続中



ネットワーク
切断時



ネットワーク
再接続後

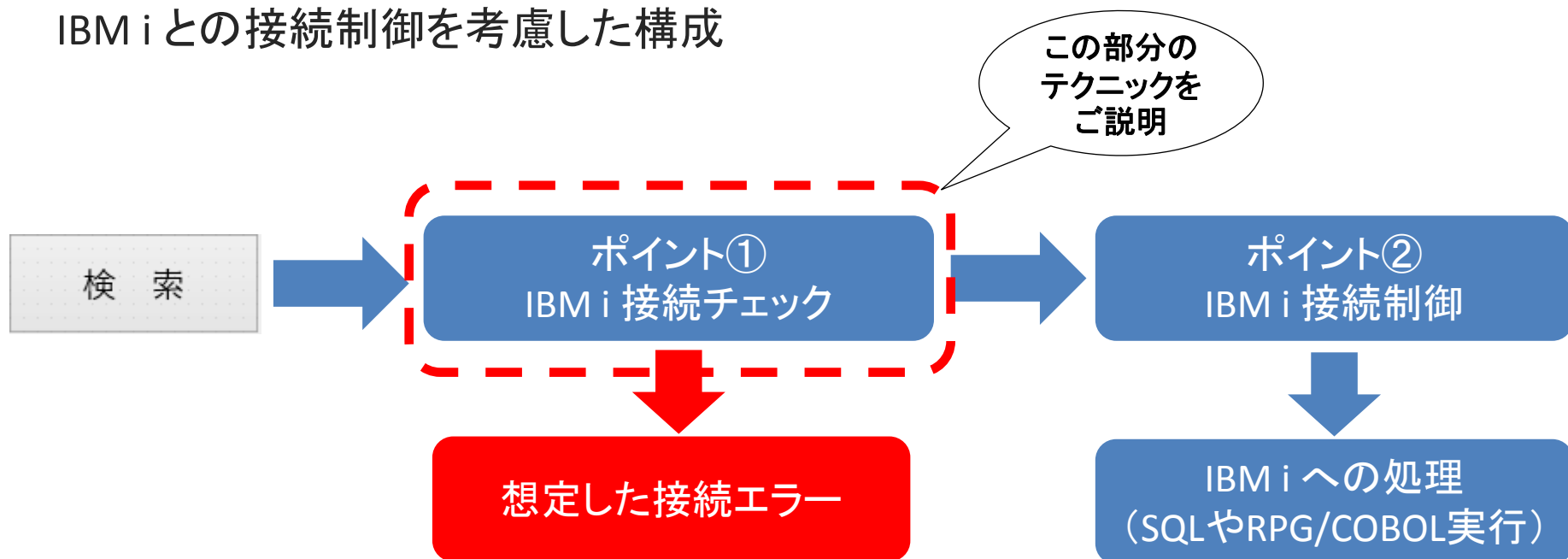


一度切断されていても
正常に処理

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック

IBM i との接続制御を考慮した構成



3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック

IBM i との接続状態をチェックする手法

【手法が有効なデバイス】

(A) 通信機能
のチェック

- iOS
- Android
- Windowsタブレット

(B) 通信経路
のチェック

- iOS
- Android
- Windowsタブレット

(C) 接続状態
のチェック

- iOS
- Android
- Windowsタブレット

インターネットに
公開されている
ライブラリを使用

Windows
APIを使用



サーバ機能
を実行

SQLを実行

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(A) 通信機能のチェック (iOS/Android) — 仕組み —

スマートデバイス自体の通信機能の有効/無効をチェックする。(iOS/Android)



デバイス内でWi-Fiなどのネットワークが有効になっているかを専用ライブラリでチェック

【通信機能チェックに使用するプログラム】

エンバカデロMVPのDave Nottageが公開している専用ライブラリ

※Dave Nottageのblogより「Network State Demo」の入手が必要。

URL <http://delphi.radsoft.com.au/2013/11/>

【ダウンロードファイルに含まれるうち、必要なファイル】

- | | |
|---------------------------|----------------------------|
| ・CaptiveNetwork.pas | ・libReachability.a |
| ・NetworkState.Android.pas | ・NetworkState.iOS.pas |
| ・NetworkState.pas | ・SCNetworkReachability.pas |

プロジェクトファイルと同じフォルダに保存

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(A) 通信機能のチェック (iOS/Android) — 実装例 —

```
uses
  NetworkState;

function TfrmMTS2015Winter.CheckDevice: Boolean;
var
  NetSta: TNetworkState;
begin
  Result := False;

  // NetworkStateの取得
  NetSta := TNetworkState.Create;
  try
    Result := NetSta.IsConnected; // ネットワーク状態が接続済みであればCheck OK
  finally
    NetSta.Free;
  end;
end;
```

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック

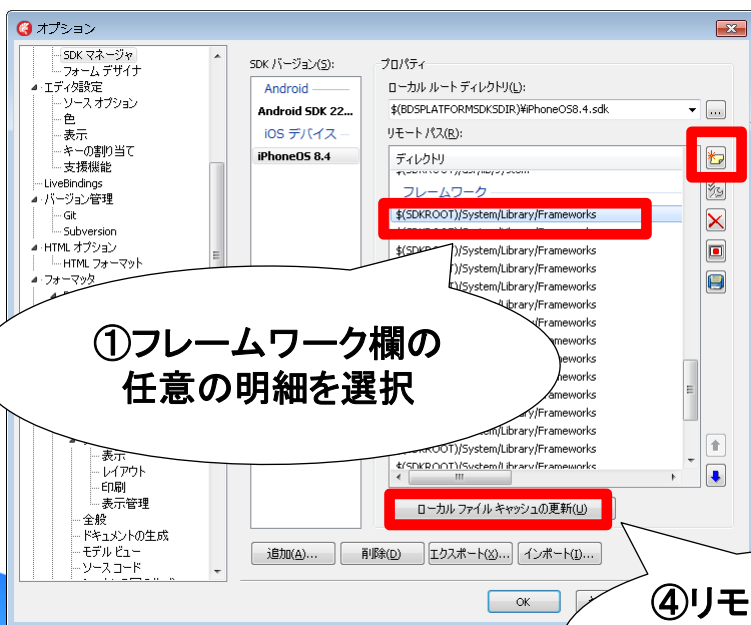


(A) 通信機能のチェック(iOS)

— 注意点 —

フレームワークの追加(iOSの場合のみ必要)

SDKマネージャでリモートパスのフレームワーク欄に項目を追加し、ローカルファイルキャッシュの更新を行う。

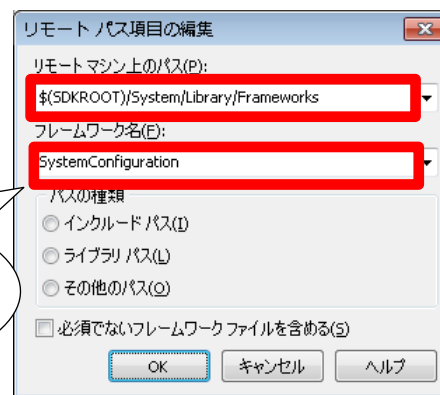


① フレームワーク欄の任意の明細を選択

② 「新しいパス項目を追加」ボタンクリック

③ 「リモートマシン上のパス」と「ファイルマスク」のみ入力

④ リモートパス追加後、ローカルキャッシュの更新を実行



リモートマシン上のパス:
\$(SDKROOT)/System/Library/Frameworks
ファイルマスク:
SystemConfiguration

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(A) 通信機能のチェック (Android)

— 注意点 —

(※NetworkState.Android.pasの24行目付近を修正する)

implementation

uses

```
System.SysUtils,  
Androidapi.JNIBridge,  
Androidapi.JNI.GraphicsContentViewText,  
Androidapi.JNI.JavaTypes,  
// FMX.Helpers.Android; // オリジナル  
Androidapi.Helpers; // XE7向け修正
```

「NetworkState.Android.pas」を
Delphi/400 XE7で使えるように、
uses節を修正する。
※XE5ではそのまま使えます。

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック

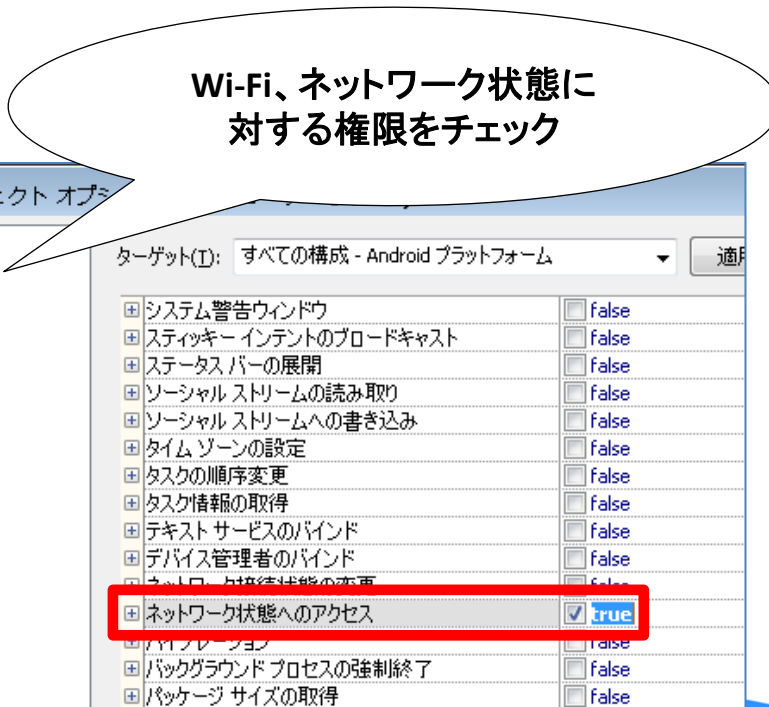
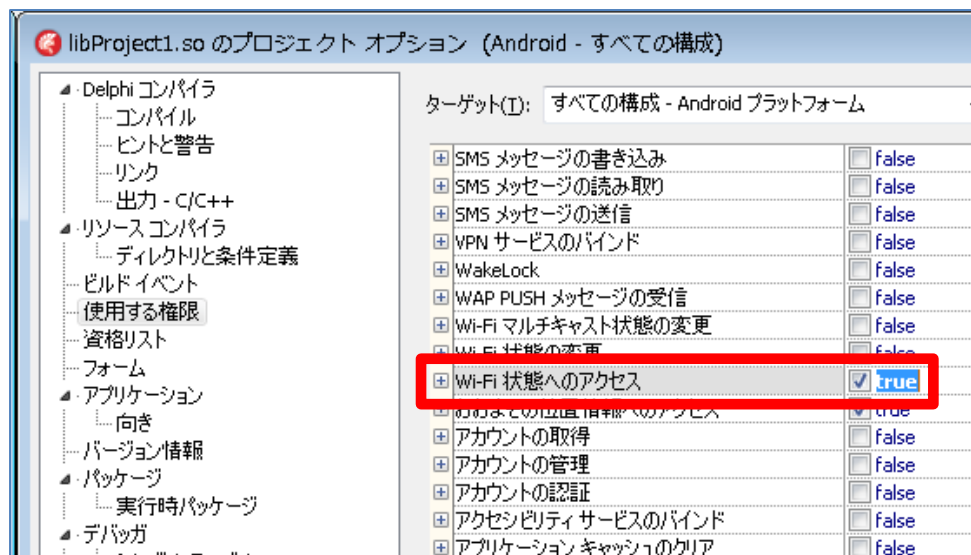


(A) 通信機能のチェック (Android)

— 注意点 —

権限設定 (Androidの場合のみ必要)

「Wi-Fi状態へのアクセス」及び「ネットワーク状態へのアクセス」権限を使用できるようにプロジェクトオプションを設定する。



3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(A) 通信機能のチェック (Windows) — 仕組み —

スマートデバイス自体の通信機能の有効/無効をチェックする。(Windows)



デバイス内でWi-Fiなどの
ネットワークが有効になっているかを
Windows APIでチェック

「Win32 インターネット拡張機能(WinInet)」を使うことで、通信機能の有効/無効をチェック可能

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(A) 通信機能のチェック (Windows) — 実装例 —

```
uses
  Winapi.WinInet;

function TfrmMTS2015Winter.CheckDevice: Boolean;
var
  Flag: DWORD;
begin
  // WinInetの関数を使った通信状態の取得
  Result := InternetGetConnectedState (@Flag, 0);
end;
```

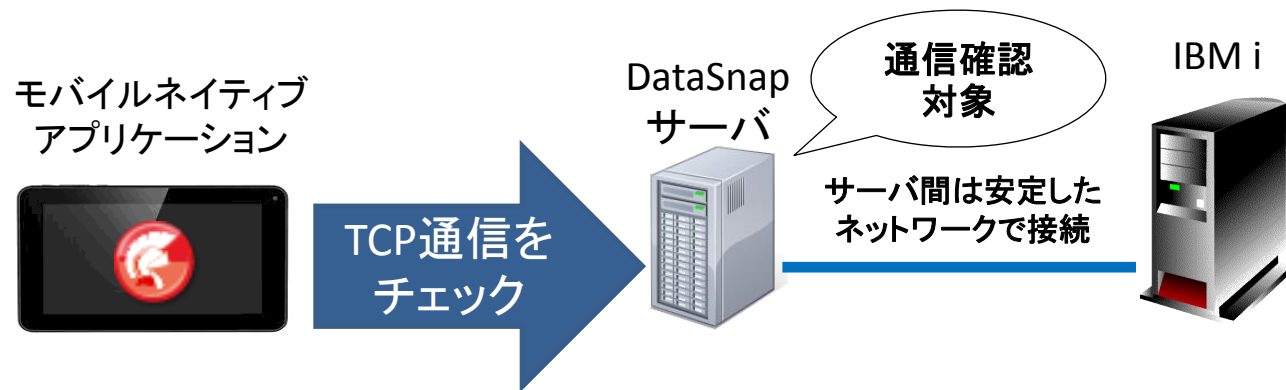
3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(B) 通信経路のチェック(共通) — 仕組み —

TIdTCPClientコンポーネントを使ってTCP通信を行い、通信経路をチェックする。



3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(B) 通信経路のチェック(共通) — 実装例 —

```
function TfrmMTS2015Winter.CheckTCP: Boolean;
begin
  Result := False;
  try
    // 接続先の指定(例: 192.168.100.200のTELNETポートへ接続)
    IdTCPClient1.Host := '192.168.100.200';
    IdTCPClient1.Port := 23;
    // 接続待ち時間の指定
    IdTCPClient1.ConnectTimeout := 2000; // 接続するまでの待ち時間(2秒)
    IdTCPClient1.ReadTimeout := 2000; // 操作完了までの待ち時間(2秒)
    // 接続確認
    IdTCPClient1.Connect; // 接続
    IdTCPClient1.Disconnect; // 切断
    Result := True; // 例外が発生しな
    // ければCheck OK

  except
    // 例外を無視する
  end;
end;
```

【ポイント】
接続待ち時間を短めに
設定する

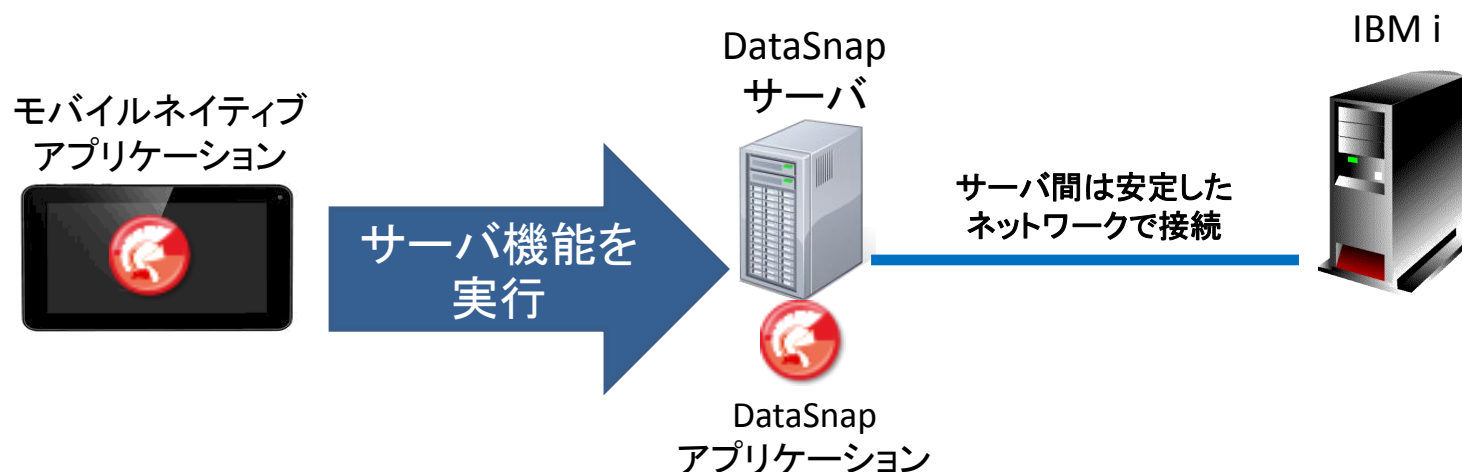
3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(C) 接続状態のチェック (iOS/Android) — 仕組み —

負荷の少ないサーバ機能を実行して、DataSnapサーバとの接続状態をチェックする。



3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(C) 接続状態のチェック (iOS/Android) — 実装例 —

```
function TClientModule1.CheckExec: Boolean;
begin
  Result := False;

  try
    // DataSnapサーバに定義した負荷の少ないメソッドを呼び出す
    // 例: DataSnap Serverのサンプルメソッド「EchoString」等
    ServerMethods1Client.EchoString('CheckSession');
    Result := True;           // 例外が発生しなければCheck OK
  except
    // 例外を無視する
  end;
end;
```

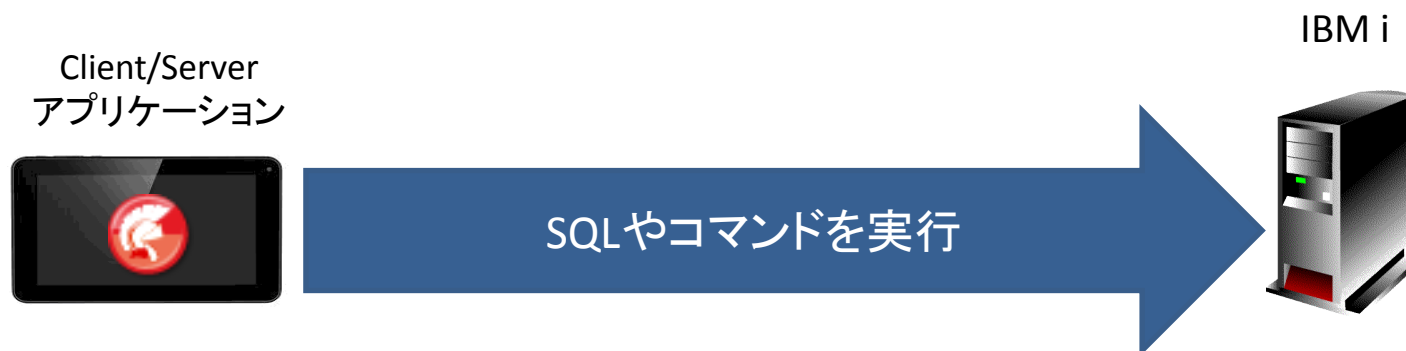
3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(C) 接続状態のチェック (Windows) — 仕組み —

負荷の少ないSQLやコマンドを実行して、IBM i の接続状態をチェックする。



3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック



(C) 接続状態のチェック (Windows) — 実装例 —

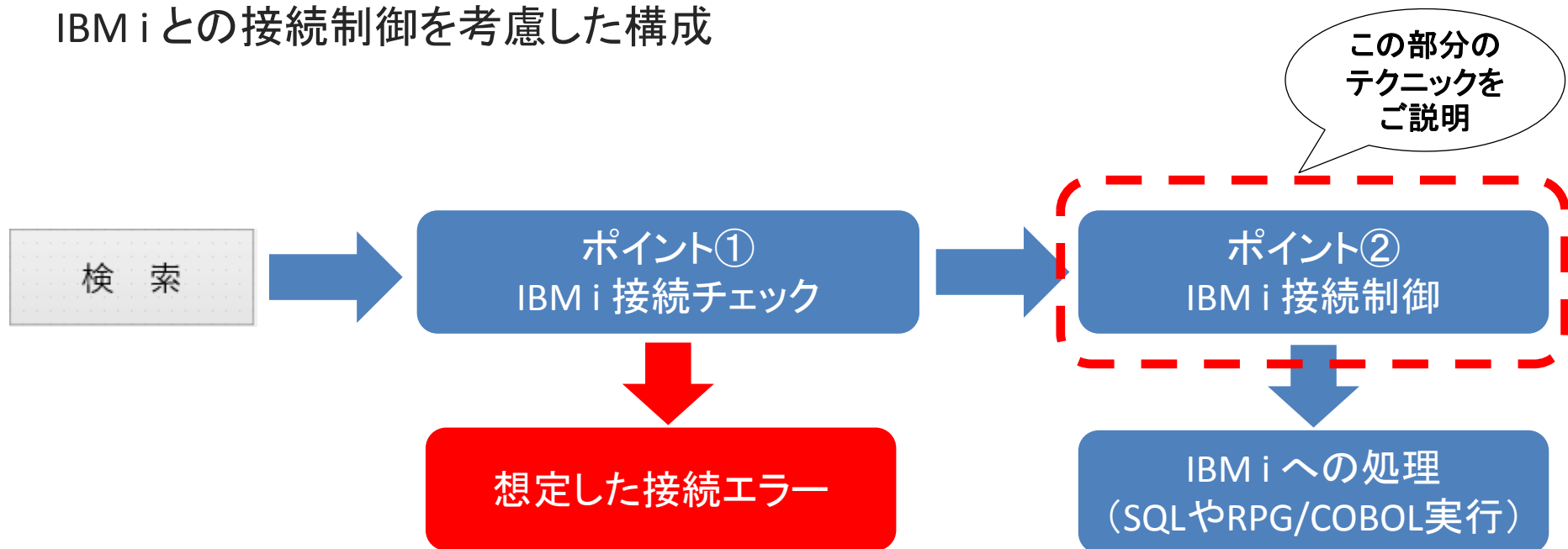
```
function TfrmMTS2015Winter.CheckExec: Boolean;
begin
  Result := False;

  try
    SQLQuery1.Active := True; // SQL (SELECT文) を実行
    SQLQuery1.Active := False;
    Result := True; // 例外が発生しなければCheck OK
  except
    // 例外を無視する
  end;
end;
```

3. ネットワーク切断を考慮したIBM i の接続制御

3-2. ポイント① IBM i 接続チェック

IBM i との接続制御を考慮した構成



3.ネットワーク切断を考慮したIBM i の接続制御

3-3.ポイント② IBM i 接続制御

IBM i と切断された場合の制御方法

【手法が有効なデバイス】

(A) 都度
接続制御

- iOS
- Android

主にモバイルネイティブ
アプリケーションで使用



(B) 再接続
制御

- Windowsタブレット

主にClient/Server
アプリケーションで使用



3. ネットワーク切断を考慮したIBM i の接続制御

3-3. ポイント② IBM i 接続制御



(A) 都度接続制御 (iOS/Android) — 仕組み —

ネットワークが一度切断された後に処理するとエラーになってしまうのは 既に切れてしまっている接続を使用することが原因。

都度接続制御では、IBM i に対する 処理毎に新たに接続して 処理を行い、処理が終わったら接続を切断する (接続チェック～切断を1セットで処理する)

モバイルネイティブ
アプリケーション



毎回新たに接続するため、切れて
しまった接続を使用することがない

接続チェック + 接続 + 検索処理 + 切断

接続チェック + 接続 + 更新処理 + 切断

IBM i



3. ネットワーク切断を考慮したIBM i の接続制御

3-3. ポイント② IBM i 接続制御



(A) 都度接続制御 (iOS/Android) — 実装例 —

```
// 主にモバイルネイティブアプリケーションの場合
begin
  // 通信経路を確認する
  if not CheckTCP then
    begin
      // 通信できない状態ならエラーとする
      raise Exception.Create('DataSnapサーバに接続できません');
    end;

  // セッションを開始する
  SQLConnection1.Connected := True;
  try
    // <必要な操作を記述>
  finally
    // セッションを終了する
    SQLConnection1.Connected := False;
  end;
end;
```

3.ネットワーク切断を考慮したIBM i の接続制御

3-3.ポイント② IBM i 接続制御



(A)都度接続制御(iOS/Android) — 留意点 —

処理毎に新たに接続するため、以前のジョブ情報やジョブに依存する機能は使えない

対処法

都度接続の場合には、毎回新たに接続するためライブラリリストの設定等を行う必要がある。
また、ジョブに依存するQTEMPやローカルデータエリアが使えないため、ジョブに依存しないメンバを使う仕組みが有効。

→モバイルネイティブアプリケーション内にもデータを保持できるため、IBM i への接続自体を減らすことが可能

3. ネットワーク切断を考慮したIBM i の接続制御

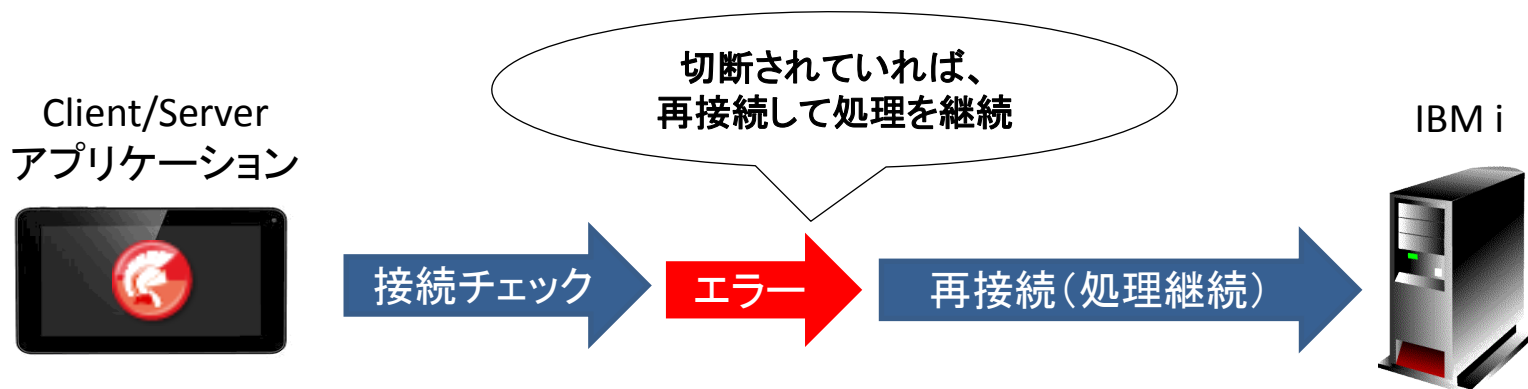
3-3. ポイント② IBM i 接続制御



(B) 再接続制御 (Windows) — 仕組み —

ネットワークが一度切断された後に処理するとエラーになってしまうのは 既に切れてしまっている接続を使用することが原因。

再接続制御では、IBM i に対する 処理実行時に切断されていたら再接続して処理を実行する。 通常は以前の接続を継続して使用する (常時接続)



3. ネットワーク切断を考慮したIBM i の接続制御

3-3. ポイント② IBM i 接続制御



(B)再接続制御 (Windows) — 実装例 —

```
// 主にClient/Serverアプリケーションの場合(1/2)
var
  i: Integer;
begin
  // 接続状態を確認する(最大3回繰り返し、4回目はエラーにする)
  for i := 1 to 4 do
    begin
      if i < 4 then
        begin
          if CheckExec then
            Break // 接続状態が維持できているなら何もせずループを終了する
          else
            begin
              try
                // 接続状態が維持できていないなら再接続する
                SQLConnection1.Connected := False;
                SQLConnection1.Connected := True;
            end;
        end;
    end;
  end;
```


3.ネットワーク切断を考慮したIBM i の接続制御

3-3.ポイント② IBM i 接続制御



(B)再接続制御 (Windows) — 実装例 —

```
// 主にClient/Serverアプリケーションの場合(2 / 2)

    except
        // 例外を無視する
    end;
end;
end
else
    // 4回目はエラーにする
    raise Exception.Create( 'IBM i に接続できません' );
end;

// <必要な操作を記述>
end;
```

3. ネットワーク切断を考慮したIBM i の接続制御

3-3. ポイント② IBM i 接続制御



(B)再接続制御 (Windows) — 留意点 —

接続を再作成した場合に、以前のジョブ情報やジョブに依存するQTEMP等の情報は残らない

対処法

再接続処理後には、初回の接続同様にライブラリリストやQTEMP等の情報を再作成する等、ジョブ環境の復旧が必要になる。
切断されることを前提とするなら、QTEMPよりメンバなどのジョブに依存しない仕組みが有効。

再接続に失敗した場合にリトライ(再実行)させることもあるが、ネットワークにつながらない環境ではリトライに意味がないため、通信経路のチェックを併用したり、リトライ回数を制限するのが妥当(リトライ回数は通常1~3回程度)

3. ネットワーク切断を考慮したIBM i の接続制御

3-4. Webアプリケーションにおける再接続の工夫(補足)



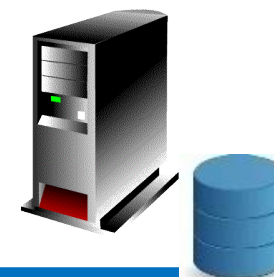
Webアプリケーションの構成の再確認

スマートデバイス上のWebブラウザから、Webサーバ上のアプリケーションを経由してIBM iへ接続し、処理を行う。

スマートデバイス

Webサーバ

IBM i



Webブラウザ



Web
アプリケーション

デバイス上に
アプリがない
||
接続制御不可能

無線ネットワーク環境に依存する接続は
Webサーバへの接続

3. ネットワーク切断を考慮したIBM i の接続制御

3-4. Webアプリケーションにおける再接続の工夫(補足)



Webアプリケーションの基本的な手法は都度接続を使用する

ポイント

スマートデバイスの接続が切れた場合、ブラウザ上ではエラーとなり再ログインが必要になるが、Webサーバ上のアプリケーションを都度接続にしておくことでWebサーバ⇔IBM i 間の処理に影響を与えない。

また、ブラウザ上の操作毎にWebサーバ経由でIBM i へ通信が必要となるWebアプリケーションはコネクションプーリングが有効である。

※コネクションプーリングについては、テクニカルレポートNo.4(2011年)の「VCL for the Web活用TIPS」も参考ください。

3. ネットワーク切断を考慮したIBM i の接続制御

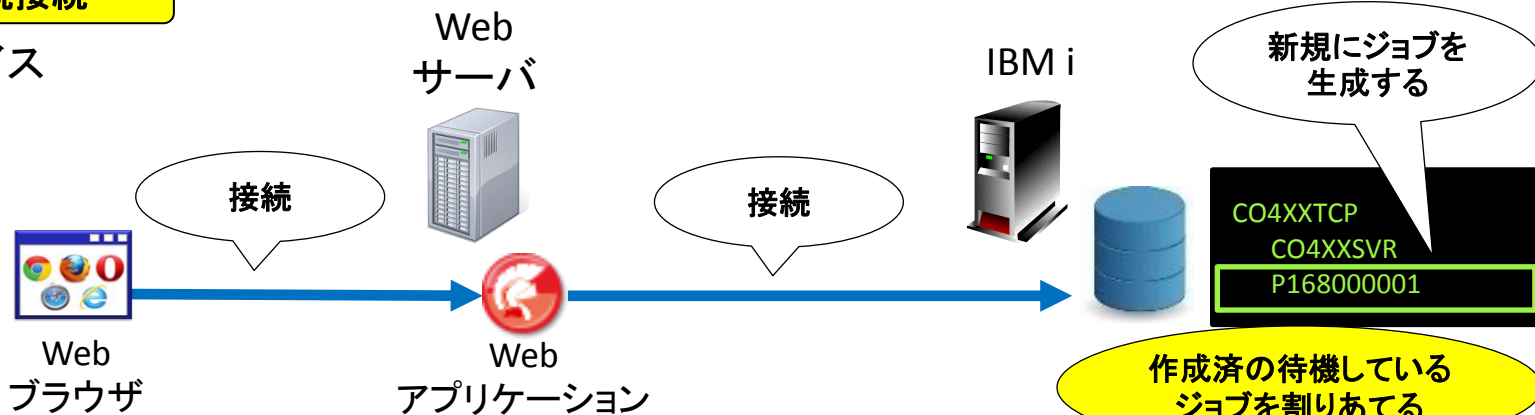
3-4. Webアプリケーションにおける再接続の工夫(補足)



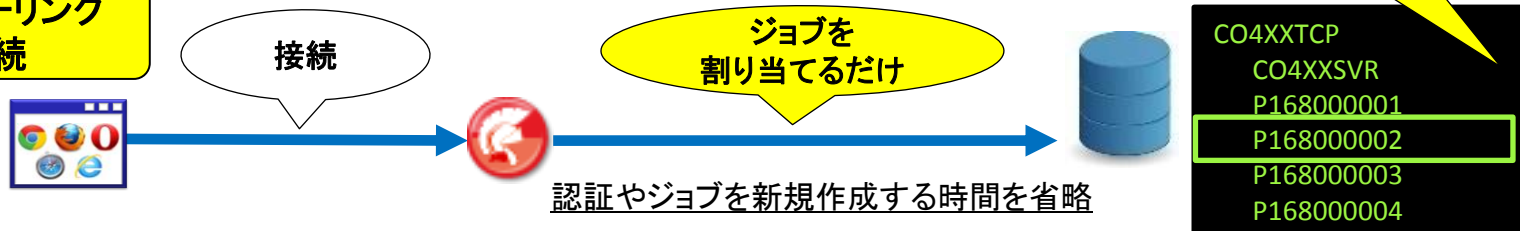
コネクションプーリング (Web) — 仕組み —

通常の新規接続

スマートデバイス



コネクションプーリング の新規接続



4.まとめ

4.まとめ

- スマートデバイスは携帯性が高く、移動中や移動先でも使用されるため、ネットワークが不安定になる場合がある
- スマートデバイスからIBM iへ接続する際の接続構成はアプリケーションの種類によって異なる
- 接続チェックはデバイス機能やTCPによる通信経路、サーバ機能の呼出等の手法がある
- 接続制御の手法は「都度接続制御」と「再接続制御」がある
- Webアプリケーションでは、通常は都度接続制御を使い、接続構成を工夫する

ご清聴ありがとうございました