

# Delphi / C++Builder / Senchaで実現する 変化に強い情報システム

第35回 エンバカデロ・デベロッパーキャンプ

エンバカデロ・テクノロジーズ  
日本法人代表 藤井 等



**e**mbarcadero®  
DEVELOPER CAMP

# 今日のトピックの出発点

## 従来型のアプリケーション

- Windows PCで使用
- RDBMSに接続
- あるいはローカルストレージにデータを保管
- 基幹業務の一端を担う

このアプリケーションに対して  
さまざまな新しい環境が関わり  
を必要としています。



# アプリケーションと外の世界のつながり

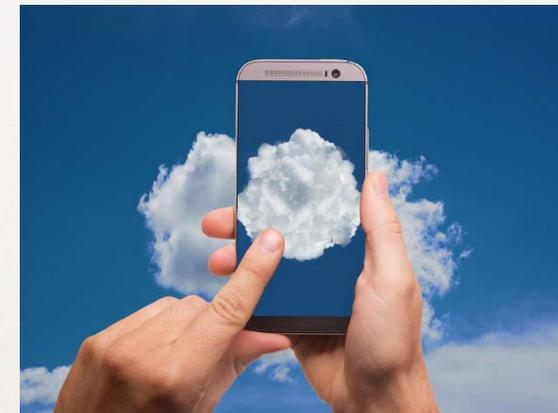
## 従来であれば...

- ファイルによるデータ交換
- 夜間バッチ
- データ転送
- :



## 現在は...

- テザリング
- アプリ間連携
- クラウド
- :



# 簡単な拡張例

- アプリケーションテザリング
  - 既存アプリケーションにテザリング機能を追加
  - モバイルデバイスと通信してデータを直接交換

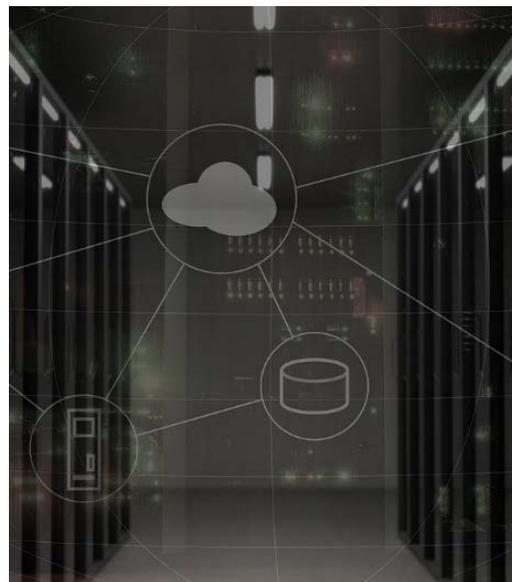


# どのような拡張があるか？

- 次のような機能が多くの企業で要求されています



モバイルデバイスの利用



クラウドサービスとの連携



Webインターフェイス



# 必要な道具は？

多様な接続性をサポートする  
フレームワーク



FireDAC



Enterprise Connectors

マルチデバイス対応の  
クライアントアプリケーション構築環境



VCL



FireMonkey



Sencha

中間層を構築するための  
ミドルウェア



RAD Server

# 同じビジュアルRADの開発パラダイムで多様な選択肢を用意

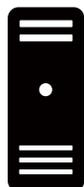
	言語	プラットフォーム	特性	接続性
 <p>VCL Windows アプリケーション</p>	Delphi C++	Windows	<ul style="list-style-type: none"> <li>Windowsのフル機能を活用可能</li> <li>Windowsデスクトップだけでなくタブレットにも対応可能</li> <li>Windows 10に適合したアプリケーションを構築可能</li> <li>旧バージョンのWindowsにも対応</li> </ul>	dbExpress FireDAC
 <p>FireMonkeyマルチデバイス アプリケーション</p>	Delphi C++	Windows macOS iOS Android	<ul style="list-style-type: none"> <li>マルチデバイス向けに単一コードベースで開発可能</li> <li>センサーなどデバイス機能の活用も可</li> <li>ローカルストレージの利用も可</li> <li>セキュアなシステム、オフラインでの利用などに最適</li> </ul>	FireDAC
 <p>Sencha Ext JS Webアプリケーション</p>	JavaScript	デスクトップ タブレット モバイル (主要なブラウザ、 プラットフォームを網羅)	<ul style="list-style-type: none"> <li>多様なブラウザで動作するモダンなWebアプリケーションを構築可能</li> <li>高機能なコンポーネントにより、従来のデスクトップアプリで提供していた機能の多くも実装可能</li> <li>スマートフォン、タブレットにも対応</li> </ul>	REST/JSON

# ビジュアルRADによって得られるメリット

- **適材適所**でツール/フレームワーク特性を活かした開発
  - いずれのフレームワークを用いても、コンポーネントによるビジュアルRADの開発パラダイムを適用可能
  - ツールの生産性、効率性を最大限発揮
- 求められる結果をすばやく獲得
  - 迅速な開発により動くアプリケーションを短時間で提供
  - 動くプロトタイプにより、ユーザーのフィードバックを得ながらの**アジャイル開発**を実践可能
- 高い品質とメンテナンス効率を実現
  - コンポーネントを用いることで、開発効率、品質、**メンテナンス効率**を劇的に向上

多様なデバイス環境に対して、目的/用途ごとに選択肢があるが  
いずれもビジュアルRADによる生産性を享受

# FireDACによる広範な接続性



 FireDAC  
データアクセスコンポーネント

-  TFDConnection
-  TFDTable
-  TFDQuery
-  TFDStoredProc
-  TFDUpdateSQL
-  TFMemTable
-  TFDBatchMove
-  TFDLocalSQL

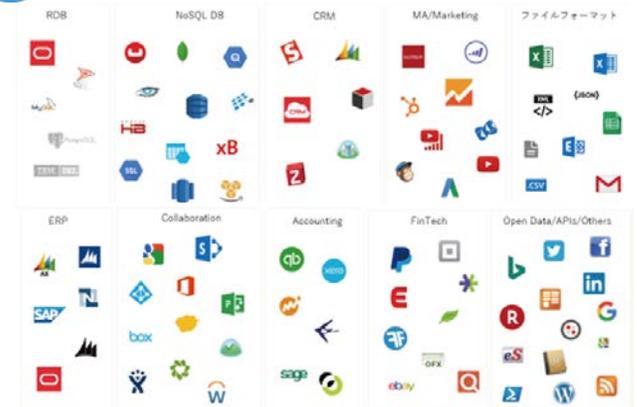
⋮

エンタープライズデータベース



and more...

EC エンタープライズサービス/クラウドサービス



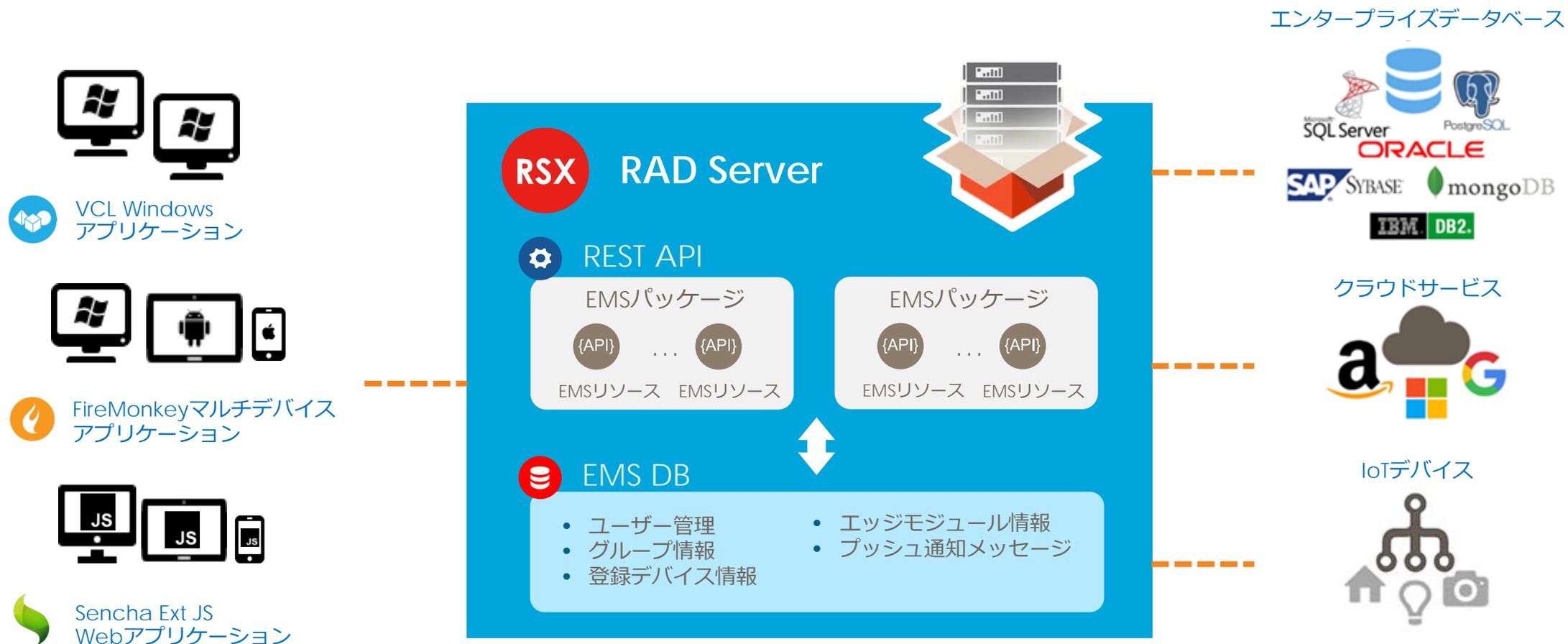
# Delphi / C++におけるFireDACの価値

- 多様なデータソースに対する**共通のアクセス性**を提供
  - 主要なエンタープライズデータベースを網羅
  - Teradata、MongoDBなどにも対応
  - 90以上のエンタープライズサービスへの接続をサポート (Enterprise Connectors)
- 従来のDelphi / C++Builderアプリケーションのデータハンドリングの方式をそのまま踏襲可能
  - TDataSet、TFieldによるデータセットの操作
  - DB対応VCLコンポーネント、LiveBindingsによるVCL / FireMonkeyコンポーネントへのデータバインド
  - サードパーティコンポーネント、レポートツールでも利用可能

FireDACを用いることで、多様なデータソースに同じ方式でアクセスできるため、データソースに対する柔軟性が生まれます



# RAD Serverで中間層を構築



# エンバカデロが考える中間層の構築

- コンポーネントによるビジュアル開発
  - サーバーサイドの開発においても**非ビジュアルのコンポーネント**により構築手順を効率化
  - FireDACおよびリモート接続用のコンポーネントを利用可能
  - データモジュールを使ってドラッグ&ドロップ開発
  - コーディング量を最小化しメンテナンス効率をアップ
- フレームワークの機能を利用し配置を容易に
  - EMSパッケージを配置するだけでREST APIとして公開可能
  - コネクションプール、認証、ユーザー管理、デバイス管理などの**エンタープライズ機能**を即利用可能



TFDDTableAdapter



TFDDSchemaAdapter



TFDDBatchMoveJSONWriter

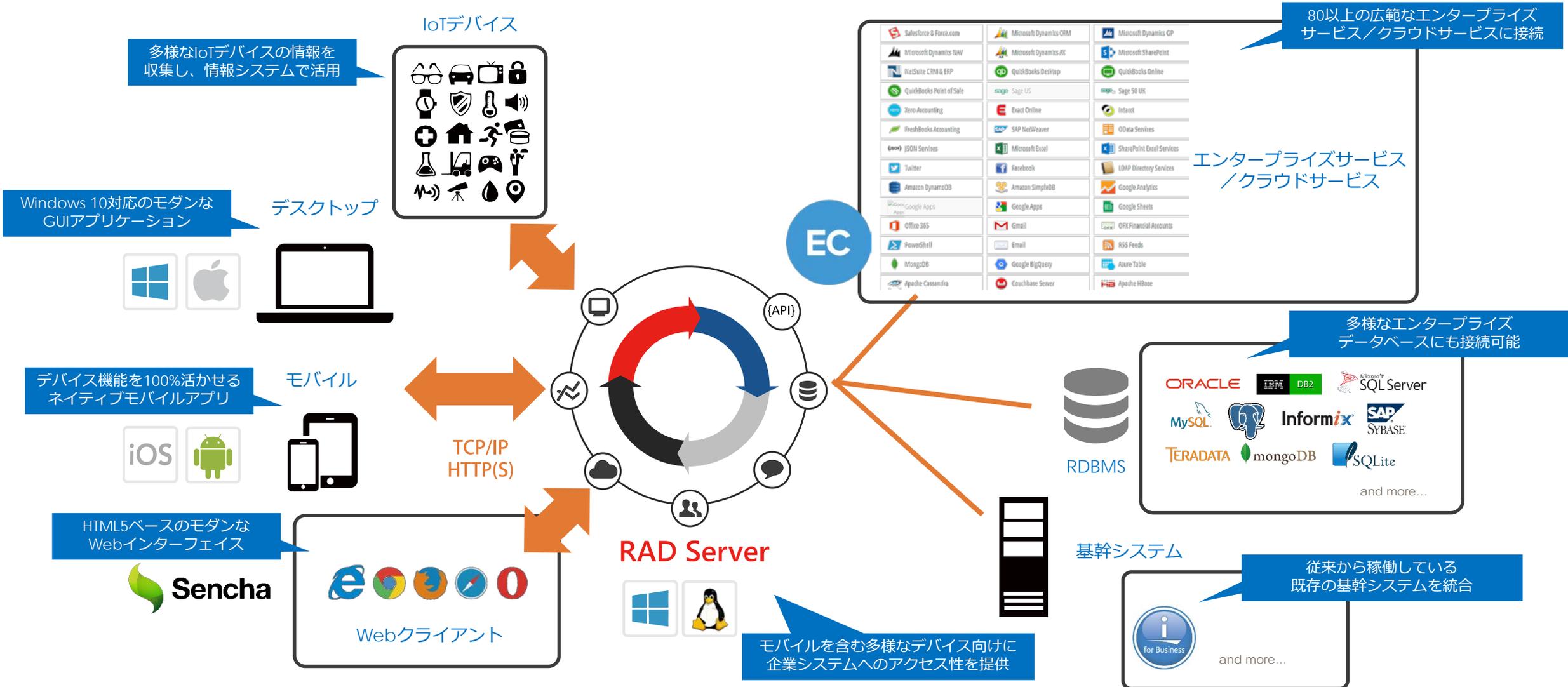


TFDDStanStorageJSONLink

:

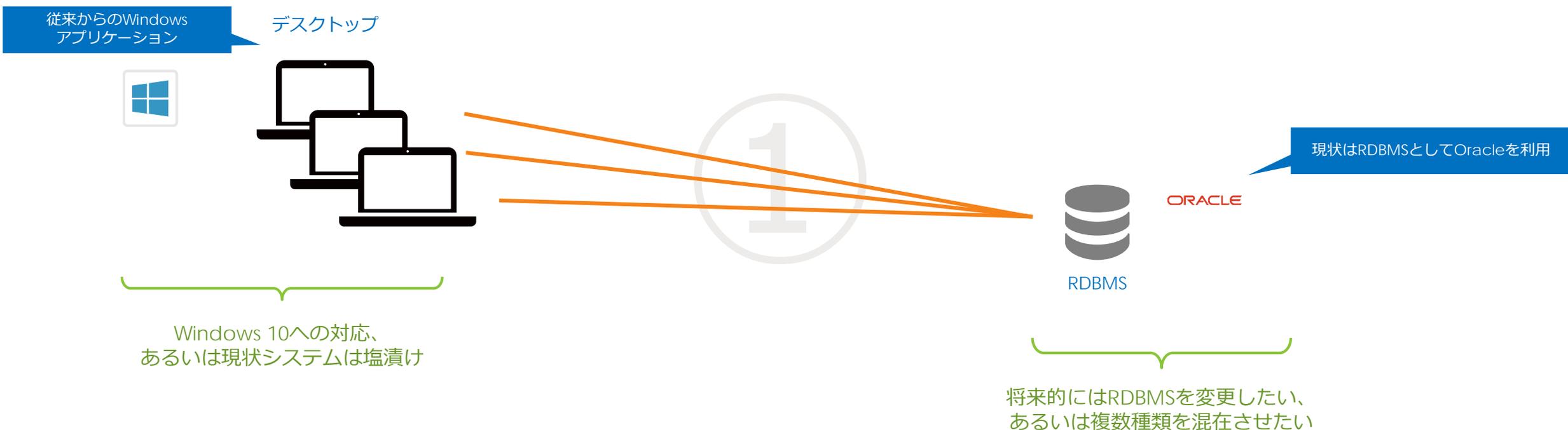
サーバーサイドで必要とされる機能を非ビジュアルコンポーネントとして提供することで、ビジュアル開発のメリットを享受可能に

# Delphi / C++Builder + Senchaで実現できること



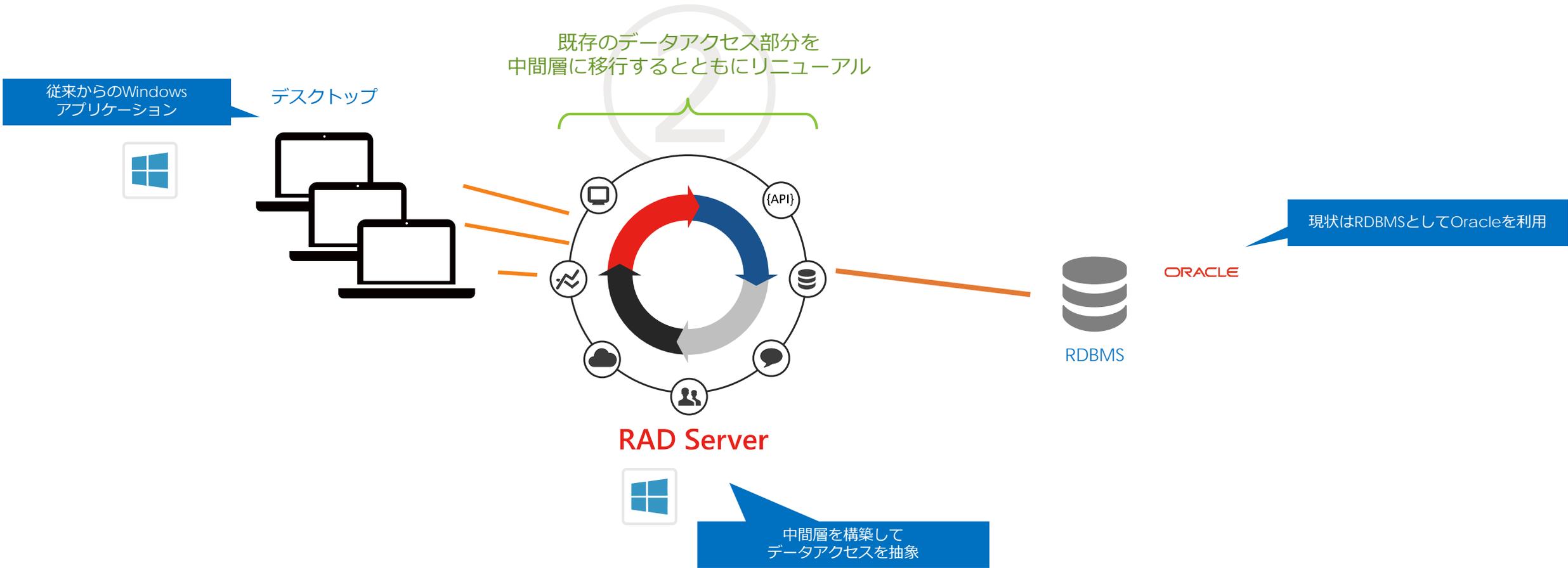
# 時系列で配置してみると...

## フェーズ1：現状のシステムと改修検討のためのロードマップ



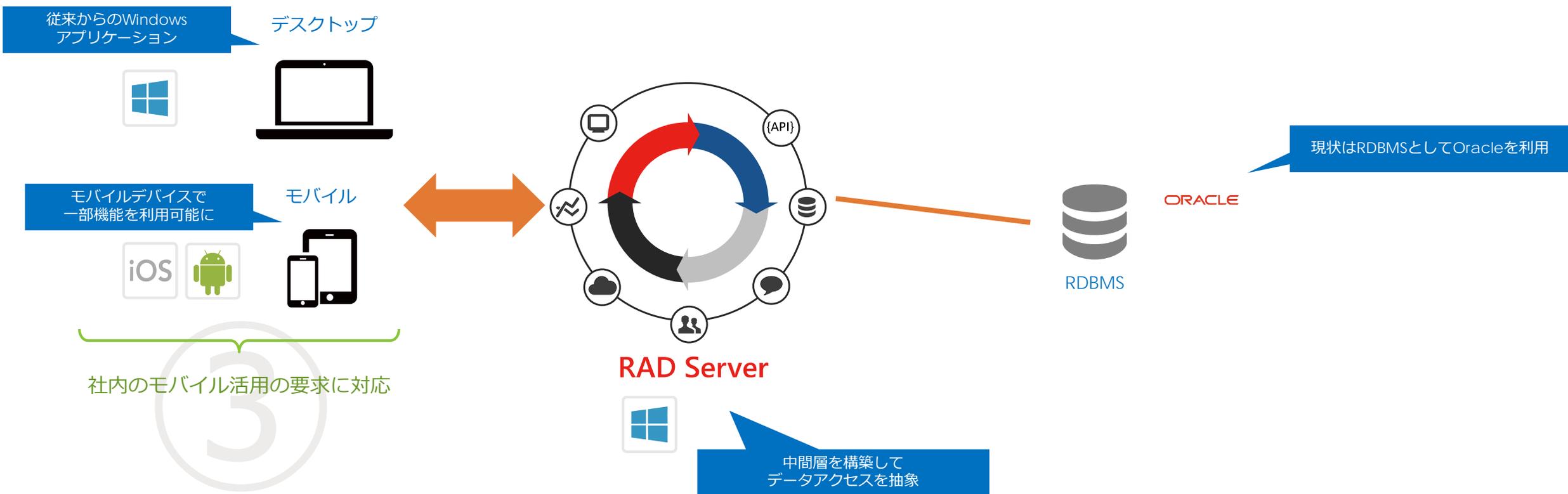
# 時系列で配置してみると...

## フェーズ2：中間サーバー層の構築



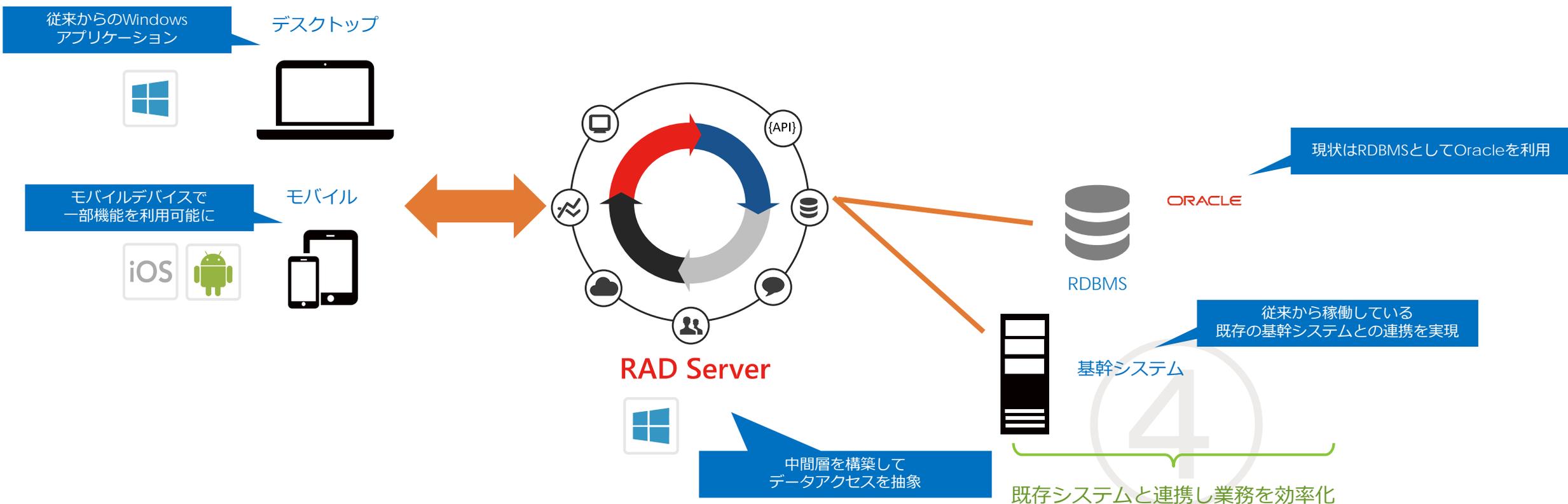
# 時系列で配置してみると...

## フェーズ3：モバイルの導入



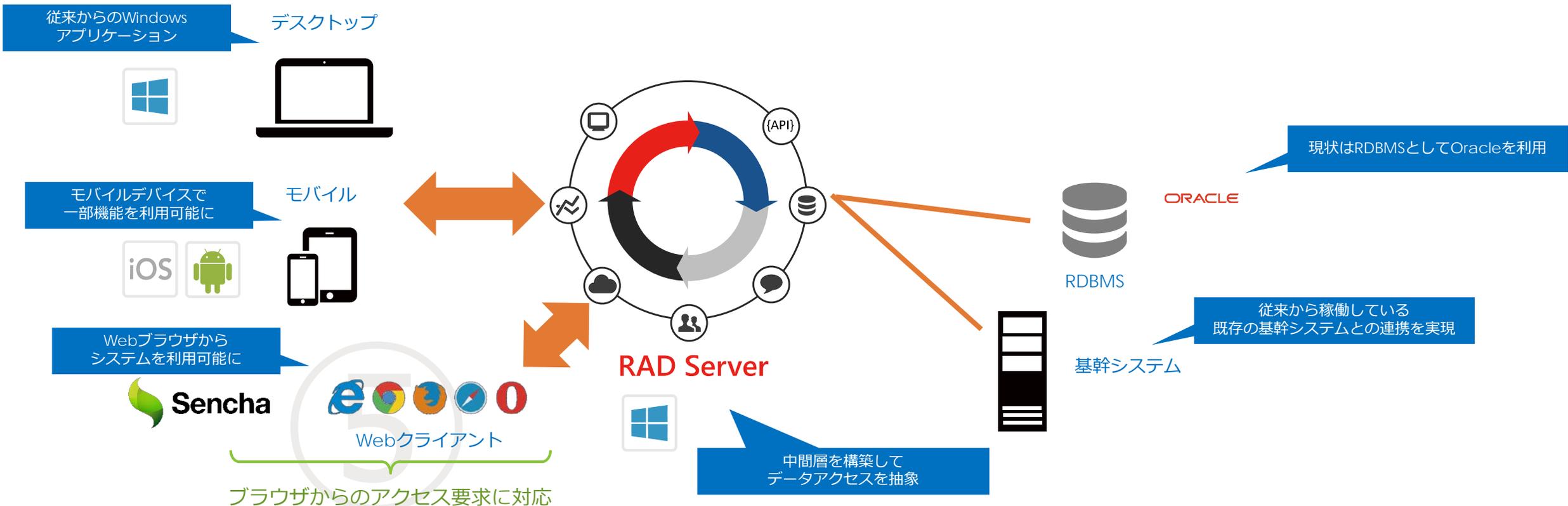
# 時系列で配置してみると...

## フェーズ4：社内の既存システム資産との連携



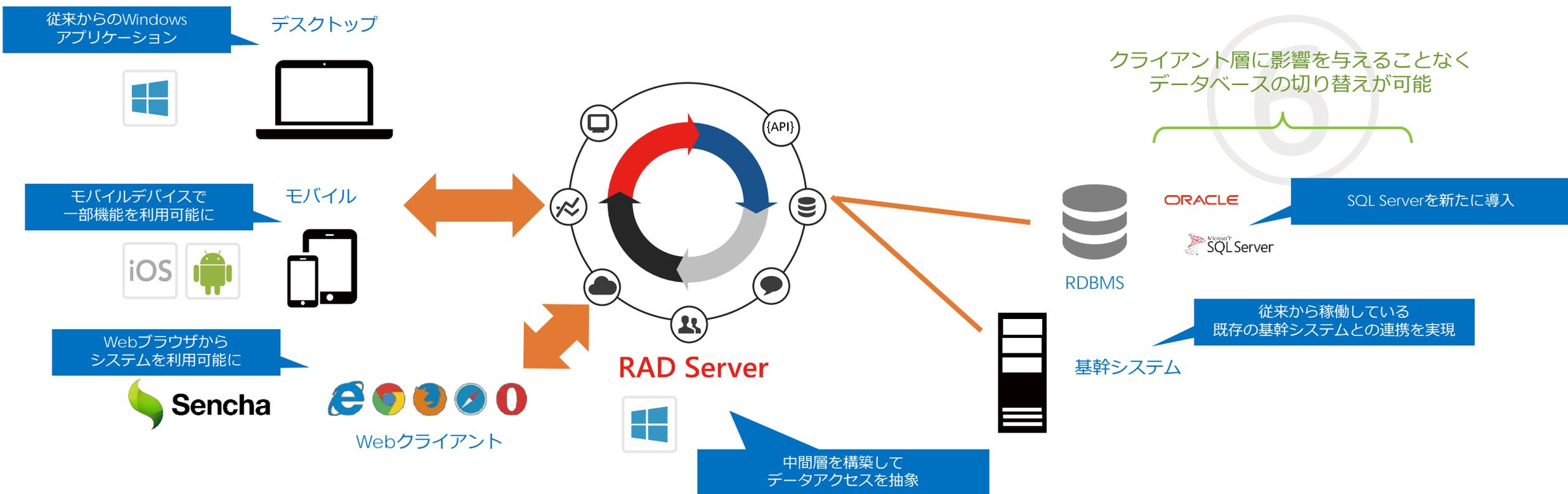
# 時系列で配置してみると...

## フェーズ5 : Webアプリケーションの導入



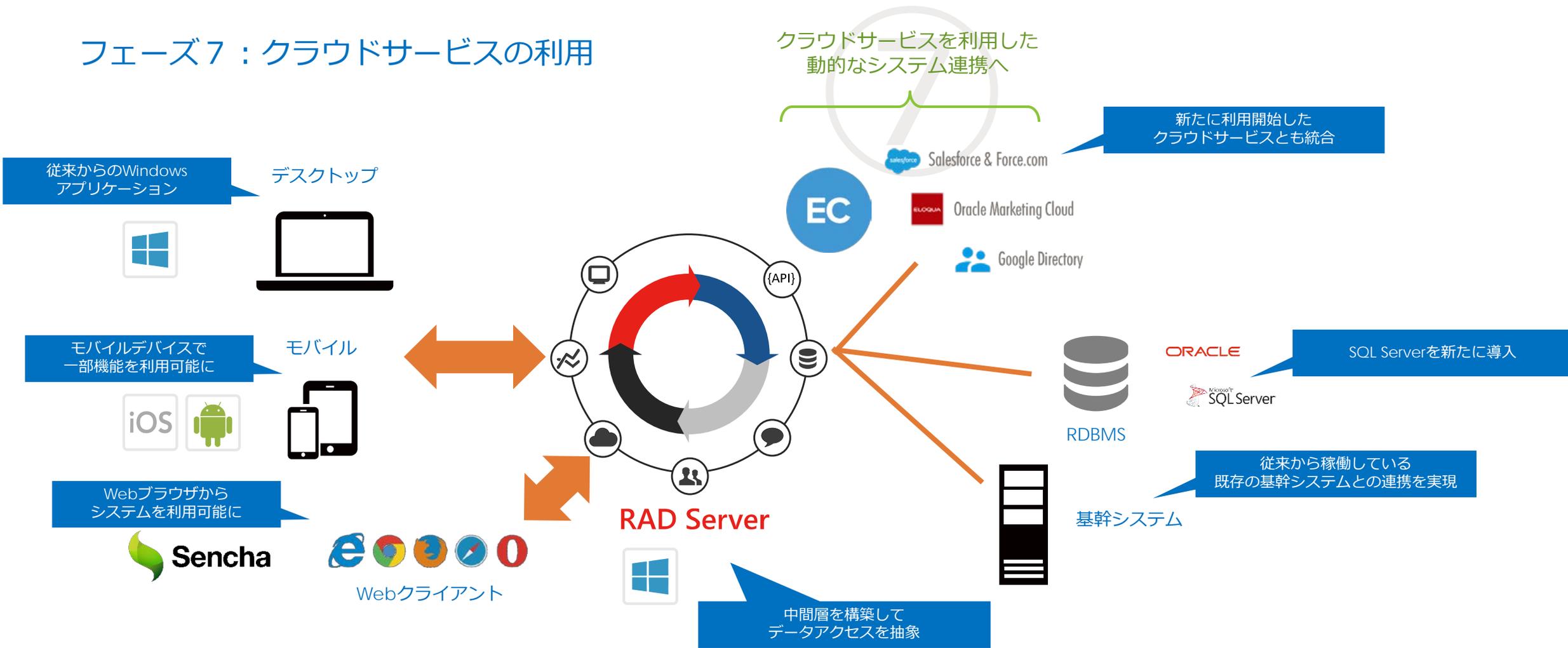
# 時系列で配置してみると...

## フェーズ6 : RDBMSの変更



# 時系列で配置してみると...

## フェーズ7：クラウドサービスの利用



# クライアントサイドで得られる柔軟性

- 適材適所のアプリケーション構築技術の適用
  - Delphi / C++を用いるべきところと、そうでないところの切り分け、割り切りが容易に
  - ある部分に異なる言語を採用しても、すべてを切り替える必要なし
- 既存システム資産を活かした拡張、改良が可能
  - 「そのまま残す（塩漬け）」という選択から、「残しながら活かす（モダナイズ）」という選択へ
  - 従来からのアーキテクチャに新しい技術を付加するかたちでの段階的な拡張が可能

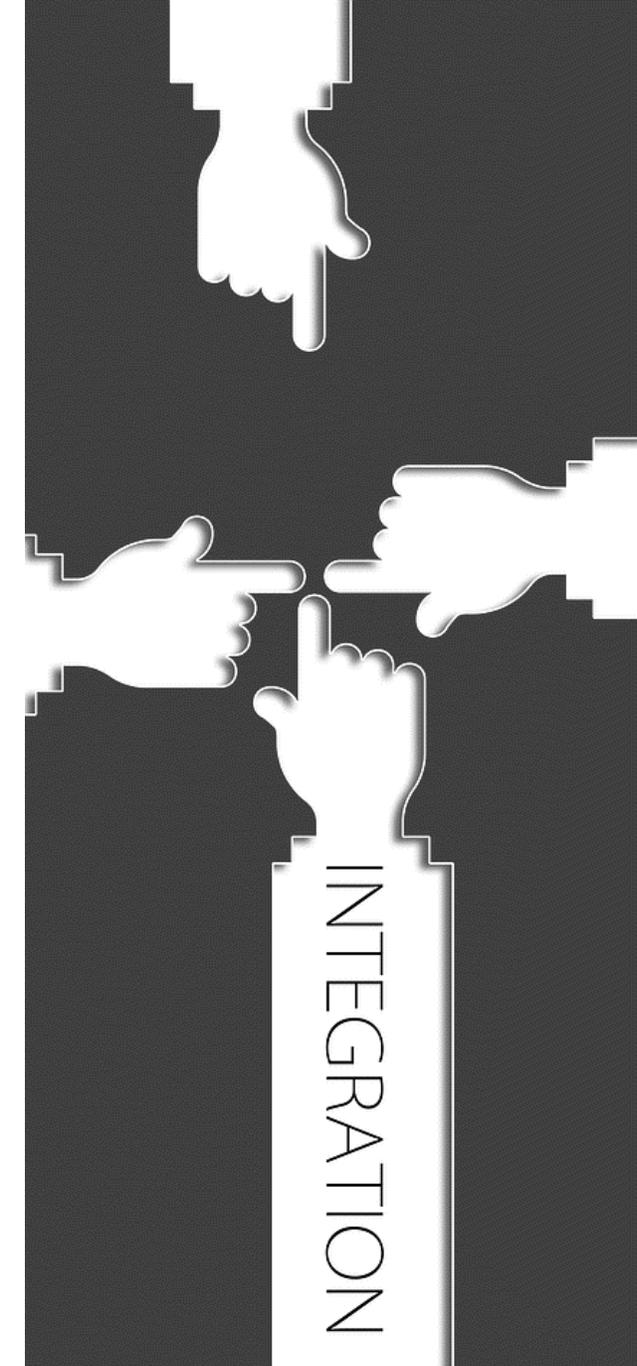


既存アプリケーション資産やスキルを活かしながら  
最新技術やトレンドの要求に対応していくことができます

# サーバーサイドで得られる柔軟性

- 任意のクライアントへ接続可能
  - Delphi、C++、Senchaに加え、Java、.NET、他のWeb技術など
  - 標準の接続方式で「機能」を利用可能な状態に
- 多様なシステムに接続できるハブとしての役割
  - レガシーシステム、クラウドなど多様なシステムをつなぐ役割
- バックエンドシステムの変更に柔軟に対応可能
  - クライアントには抽象化された「機能」を公開
  - どのようにそれを実現しているかはブラックボックス化が可能
  - サーバーサイドの構成や設置場所にも柔軟性が

移行プロジェクトや既存システム資産の段階的な  
リプレースにも有効な手段となります



# BUILDING A FUTURE TOGETHER

## 未来をともに...

エンバカデロでは「過去を捨てずに未来へ」をモットーに、開発者の皆さんに新しい技術をキャッチアップできる環境の提供に努めています。皆さんのシステム構築を、より円滑に将来にわたってサポートできることを願っています。



# THANKS!

[www.embarcadero.com/jp](http://www.embarcadero.com/jp)

第35回 エンバカデロ・デベロッパーキャンプ