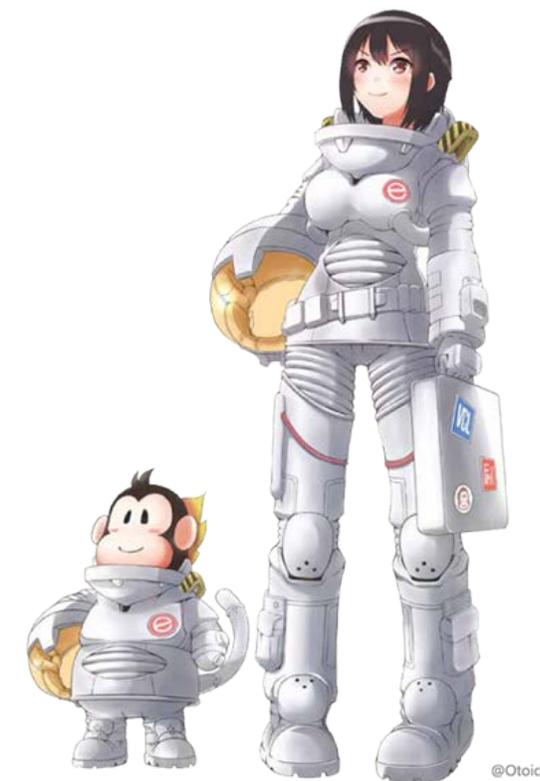


基幹系だけじゃない！シーン別データベース活用術 - モバイル、デスクトップから多層、クラウドまで、コスト節約と性能実現のキモ

第34回 エンバカデロ・デベロッパークャンプ

エンバカデロ・テクノロジーズ
セールスコンサルタント
井之上 和弘



embarcadero®
DEVELOPER CAMP

はじめに

- データベースといえは、バックエンドのエンタープライズDBのみを想像しがちですが、クライアントや中間層でもデータの格納は必要です。
- デスクトップ、モバイルデバイス、中間サーバーなど、さまざまなシーンでどのようにデータを格納し、システム全体の性能アップ、利便性の向上を図るかを、Delphi / C++BuilderとInterBaseを活用した具体例をベースに解説します。

BDEはもう使えない？

- BDEと言えば、Delphi6までの標準ツールでした。現在も入手可能ではありますがサポートは終了しています。
- BDEの代わりに、今後は何を使えばよいのでしょうか？



BDEはもう使えない？ - 課題

- 正式サポートOSはWindowsXPまで
 - Windows7/8.1/10 での動作保証はありません
- Unicode使えません
 - AnsiString (ShiftJIS) だけです
- 64bitアプリでは使えません
 - BDEは32bit版だけです
- マルチプロセッサをサポートしていません
 - マルチプロセッサなPCで動かすとエラーが出たりフリーズしたり
- Windows Vista以降のOSではBDE利用のアプリ実行時に制約が出ることも
 - 管理者権限が必要となる場合があるなど

BDEはもう使えない？ – 解決策・提案

■ TDataSet, TField, TDBGrid 等はFireDACで利用できます

- Delphi/C++Builder XE3以降ではFireDACが標準のDBアクセス用フレームワークです
- BDEで使用してきた TQuery, TTable の代わりに TFDQuery, TFDTable を使います。

■ Delphi/C++Builderで利用可能な組み込みDBを利用

- InterBase(IBLite, IB ToGo), SQLite などの**組み込みDB**がFireDACで利用できます
- 組み込みDBはライブラリとしてアプリに含まれるので**別途のインストールが不要**
- IBLiteは**容量100MB以下、暗号化無し**の場合に利用できます
- 暗号化データベースを利用したい場合は InterBase ToGo

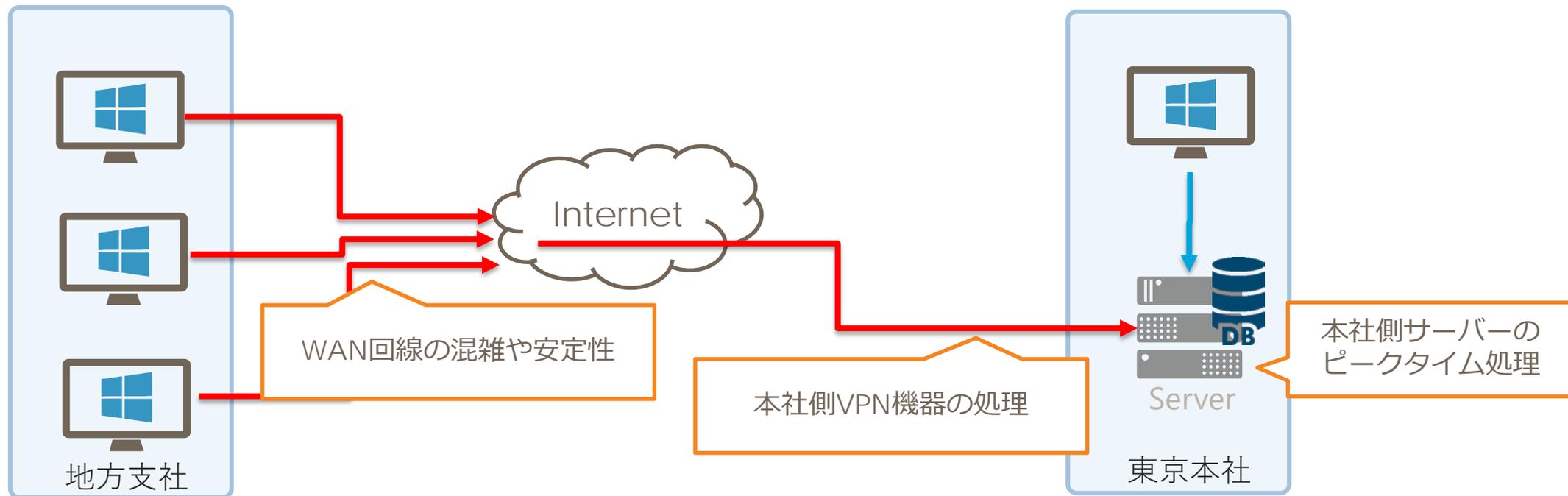
■ 低レスポンスな拠点間接続の憂鬱

- 複数の拠点を構えるオフィス環境では、オンプレミスな社内情報資産へのアクセス速度が利用者の利便性を損なう場合があります...



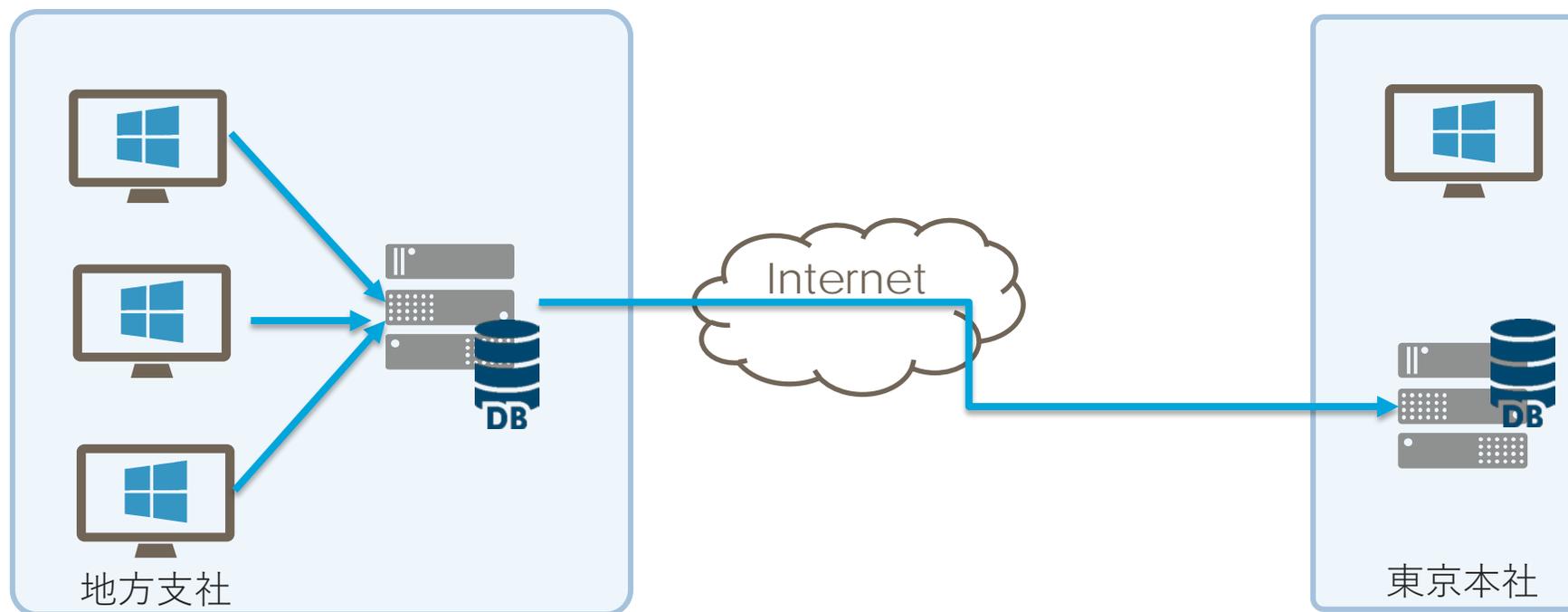
低レスポンスな拠点間接続の憂鬱 - 課題

- WANやVPNで本社サーバーに接続するとレスポンスが遅くて困ることが.....



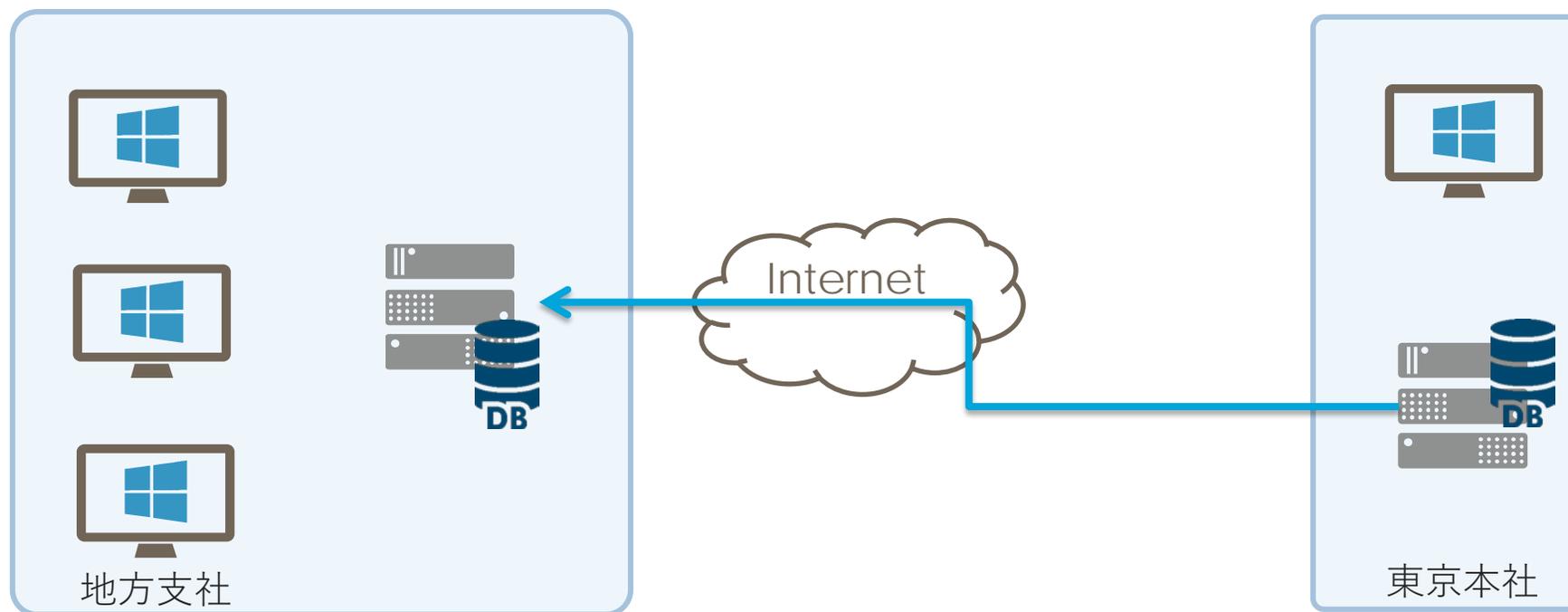
低レスポンスな拠点間接続の憂鬱 - アイデア？

- 地方拠点にデータベースのコピーまたはキャッシュを持ってないだろうか？



低レスポンスな拠点間接続の憂鬱 - アイデア？

- オリジナルのDBで更新されたデータを容易に知ることはできないだろうか？



低レスポンスな拠点間接続の憂鬱 – 解決策・提案

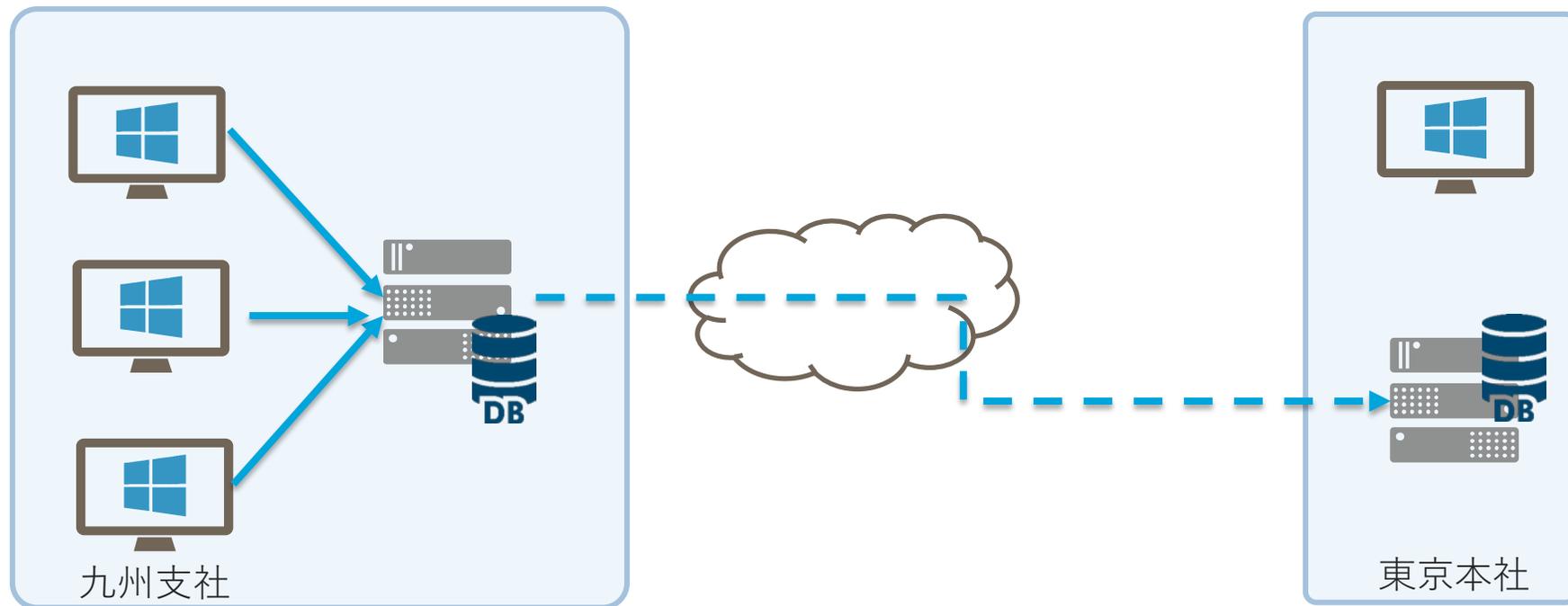
- FireDAC では TFDEventAlerter でDBの状態の変更通知を受けられる
 - ただし DB によってサポートする Alert の種類は異なることに注意
- InterBase では “更新ビュー” 機能を使えば、更新情報を容易に取得できる

[デモ]

- TFDEventAlerter
 - C:\Users\Public\Documents\Embarcadero\Studio\19.0\Samples\Object Pascal\Database\FireDAC\Samples\DBMS Specific\InterBase\ChangeView
- ChangeView
 - FDEventAlerterでを使用したDBに対して、変更ビューを設定して更新部分だけが取得できることを見せる
 - CodeRage 9 の InterBase Keynote の内容の抜粋
 - <https://www.youtube.com/watch?list=PLwUPJvR9mZHhgQYWWgF6BCkgDwlpP78ZV&v=Vzm-ISPuKLO>
 - <http://docwiki.embarcadero.com/InterBase/2017/ja/変更ビューにサブスクリプションを作成する>

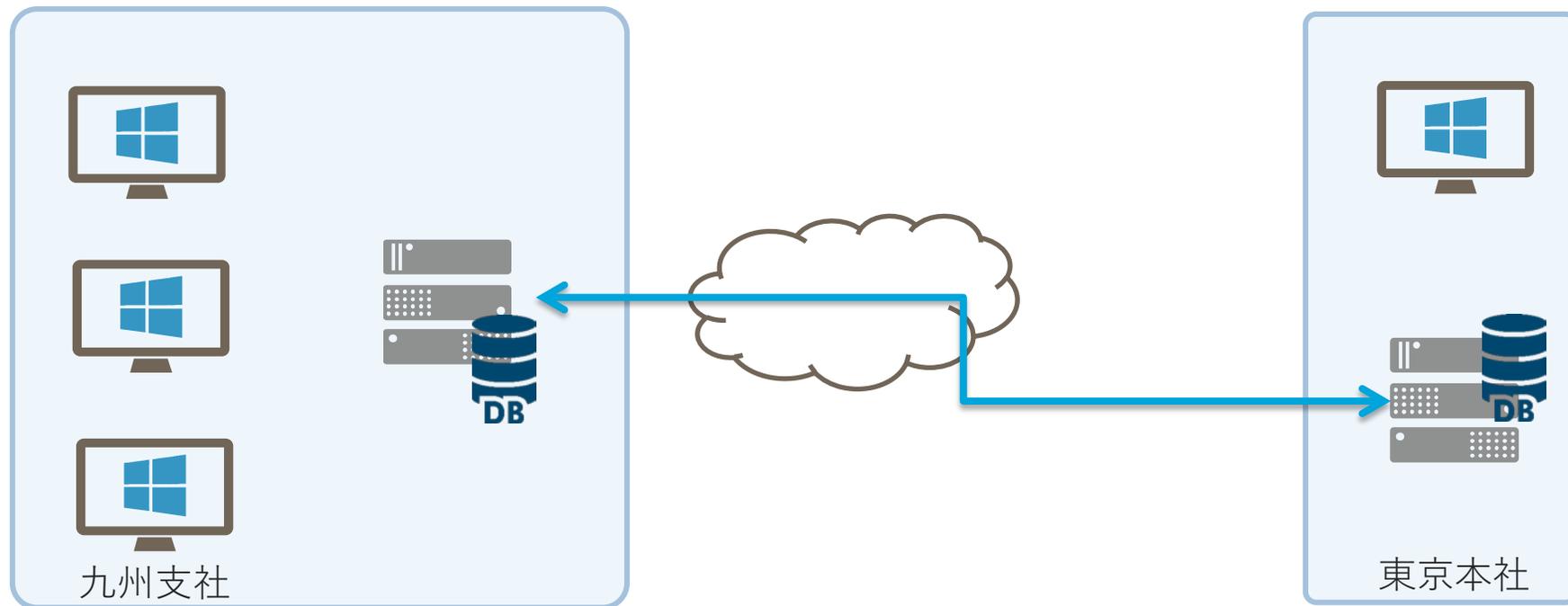
低レスポンスな拠点間接続の憂鬱 – 解決策・提案(2)

- 参照専用のデータは各拠点のキャッシュサーバーにコピーを配置すれば、本社のサーバーへのアクセスを低減できる



低レスポンスな拠点間接続の憂鬱 – 解決策・提案(2)

- TFDEventAlerterや更新ビュー機能で検知された変更分を、拠点のデータベースサーバーに反映させることが可能



低レスポンスな拠点間接続の憂鬱 – 解決策・提案(3)

- DBサーバーは便利だが、運用コストや管理を考えると……
 - 常勤の管理者による管理が難しい場合は、ゼロアドミニストレーションなデータベースの利用を検討しましょう
 - InterBase Server など

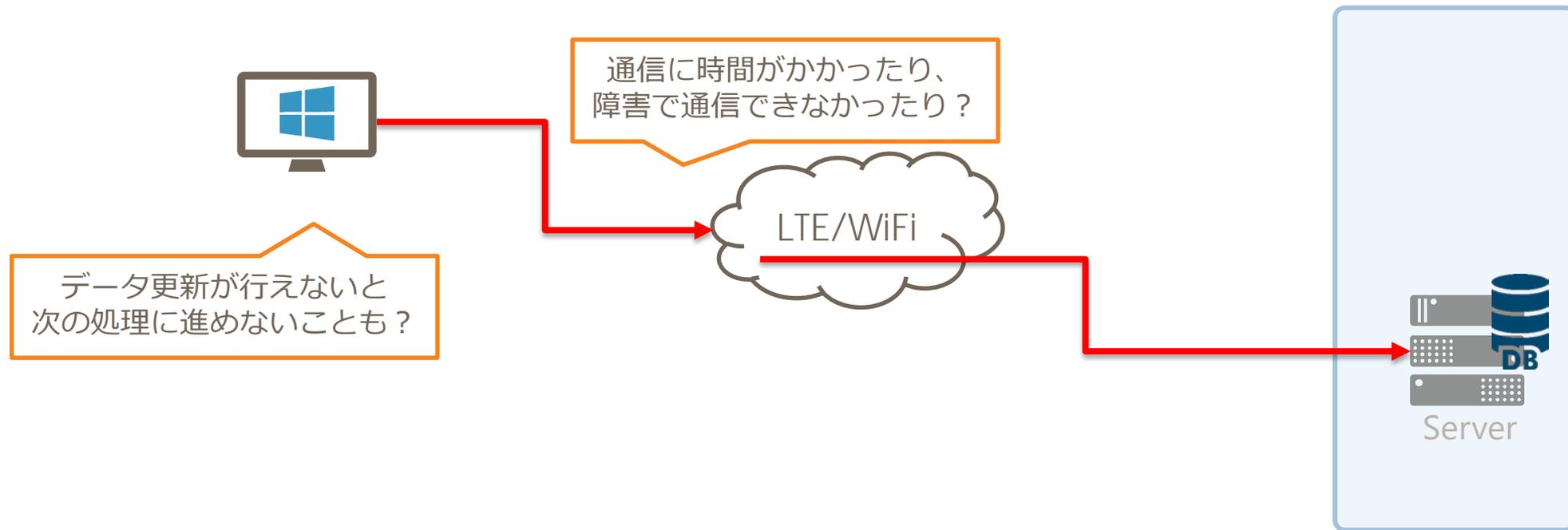
回線が不安定な場所からのデータ送信を快適に

- 社外でノートパソコンやモバイル端末で業務を行うことは増えていますが、その際に使用するモバイル回線は場所や時間帯によって状態が変わります
- 状態によらず、快適かつ安全に使うには？



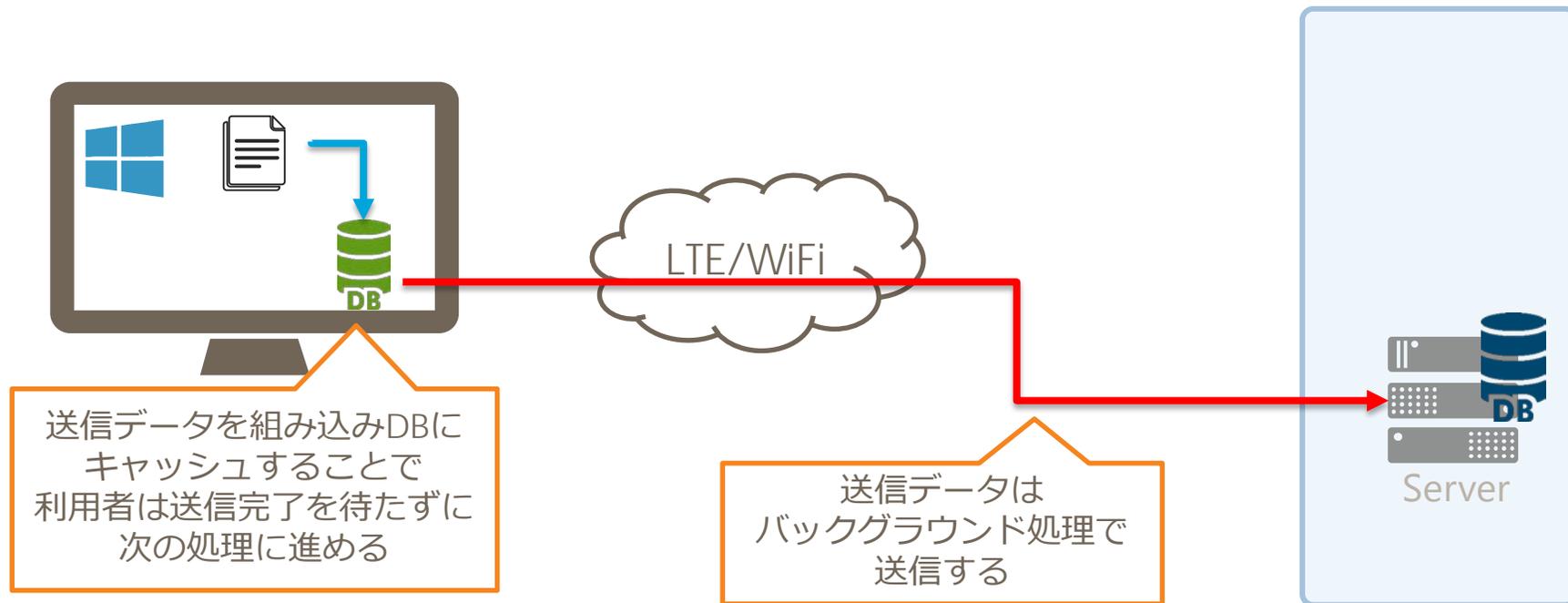
回線が不安定な場所からのデータ送信を快適に - 課題

- モバイル環境でサーバーにデータを送信するアプリは、接続が不安定だと作業が滞ります



回線が不安定な場所からのデータ送信を快適に – 解決策・提案(1)

- サーバーに直接送信するのではなく必ずローカルDBに書き込む
 - サーバーへの送信は別のプロセスやスレッドで行う



回線が不安定な場所からのデータ送信を快適に – 解決策・提案(2)

- FireDAC接続ならCachedUpdatesを使う
 - CachedUpdate を有効化すると、以後のDBへの操作がキャッシュされる。
 - ApplyUpdate でDBへの反映を行う。

```
FDQuery1.CachedUpdates := True;  
...  
FDQuery1.Edit;  
...  
FDQuery1.Post;  
...  
FDQuery1.Append;  
...  
FDQuery1.Post;  
...  
FDQuery1.ApplyUpdates;  
FDQuery1.CommitUpdates;  
FDQuery1.CachedUpdates := False;
```

see also: [http://docwiki.embarcadero.com/RADStudio/Tokyo/ja/更新のキャッシング\(FireDAC\)](http://docwiki.embarcadero.com/RADStudio/Tokyo/ja/更新のキャッシング(FireDAC))

回線が不安定な場所からのデータ送信を快適に – 解決策・提案

- 大事なデータは組み込みデータベースを使う
 - 実装工数は少し増えるが安全性は高い
- FireDAC標準機能の `CachedUpdates` でオンメモリキャッシュ
 - 少々の実装の変更だけで利用できる
- サーバーとの通信処理はバックグラウンドで
 - 利用者にはデータの更新状態を意識させることなく使わせることができる

回線が不安定な場所からのデータ送信を快適に – 選択肢

- 組み込みDBは必要なデータ量、暗号化などを考慮して選ぶ

	商用サポート	データ容量	暗号化
InterBase (IBLite)	○	△	
InterBase (IB ToGo)	○	○	○
SQLite		○	△

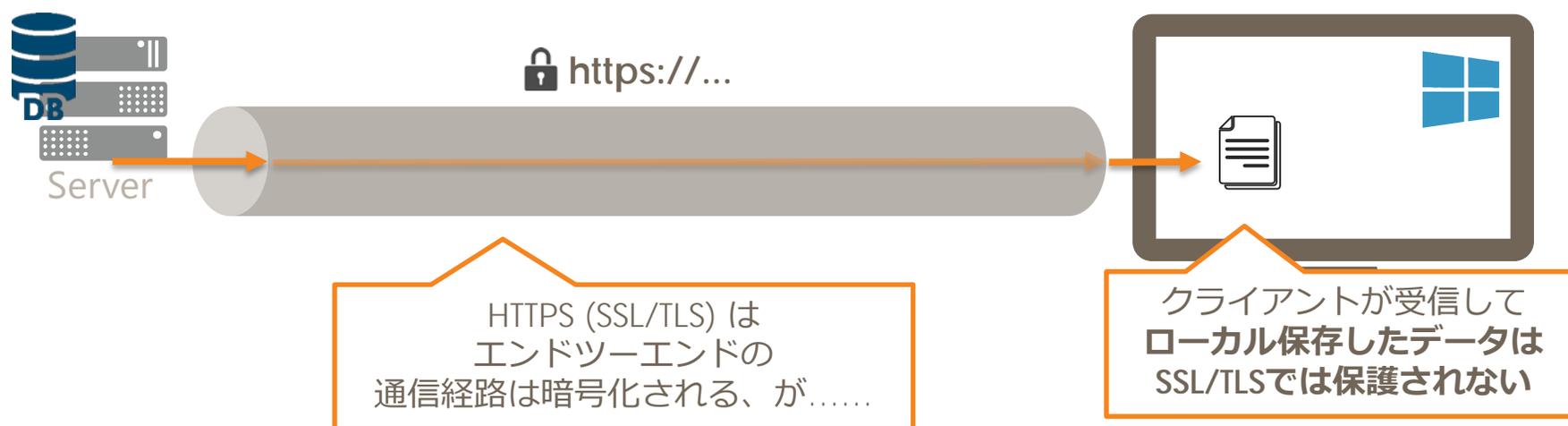
■ 関係者外秘のデータを安全に配布したい

- 従業員の社外持ち出し端末や取引先に対して、関係者外秘のデータを配布する場合に、どのような方法があるでしょうか？



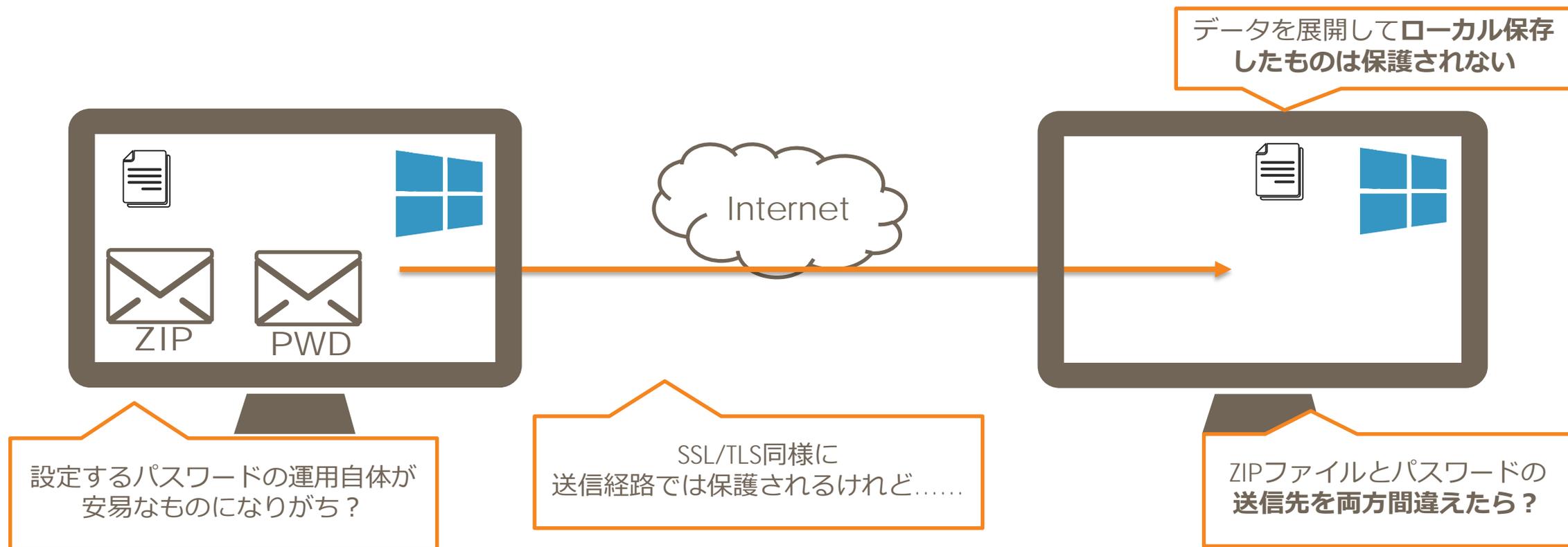
関係者外秘のデータを安全に配布したい – 課題

- 通信経路を HTTPS (SSL/TLS) にするだけで、本当に安全ですか？



関係者外秘のデータを安全に配布したい – 課題

- 「パスワードをかけたZIPファイル」で送りますか？



関係者外秘のデータを安全に配布したい – 解決策・提案

- 「業務アプリで使うデータを配布する」ことが目的ならば、そもそも「組み込みDB用の暗号化DBファイルを配布」しては？



関係者外秘のデータを安全に配布したい – 解決策・提案

- 暗号化データベースを配布するメリット？
 - データの閲覧時もストレージ上のデータが暗号化されているので、データが漏洩するリスクが低い
 - SSL/TLS や パスワード付きZIPは、受信後のデータを保護する手段ではない
- 暗号化データベースを配布するデメリット？
 - 用途は限定的である（一般的なオフィス文書やデータの配布用ではない）
- その他のメリット？
 - 暗号化パスワードを配布データ作成アプリや閲覧アプリにハードコーディングすれば、暗号化パスワードの配布は不要（ただしアプリの安全な配布方法は別途考慮が必要）

関係者外秘のデータを安全に配布したい – 選択肢

- 暗号化に対応した組み込みDBのデータファイル配布
 - InterBase ToGo など

■外部サービスのAPI利用を、より快適に

- クラウドAPIの利用では、拠点間接続同様に WAN 回線の状態によって応答時間が変わります
- WAN回線やクラウドサービスの障害によって外部サービスからデータが取得できない場合に備えて、どのような対策が取れるでしょうか



外部サービスのAPI利用を、より快適に – 課題

- 外部サービスのAPIは応答時間が一定ではない

The screenshot shows the mitmproxy interface with a table of intercepted requests. The table has columns for Path, Method, Status, Size, and Time. Three requests to the same URL are shown with different response times.

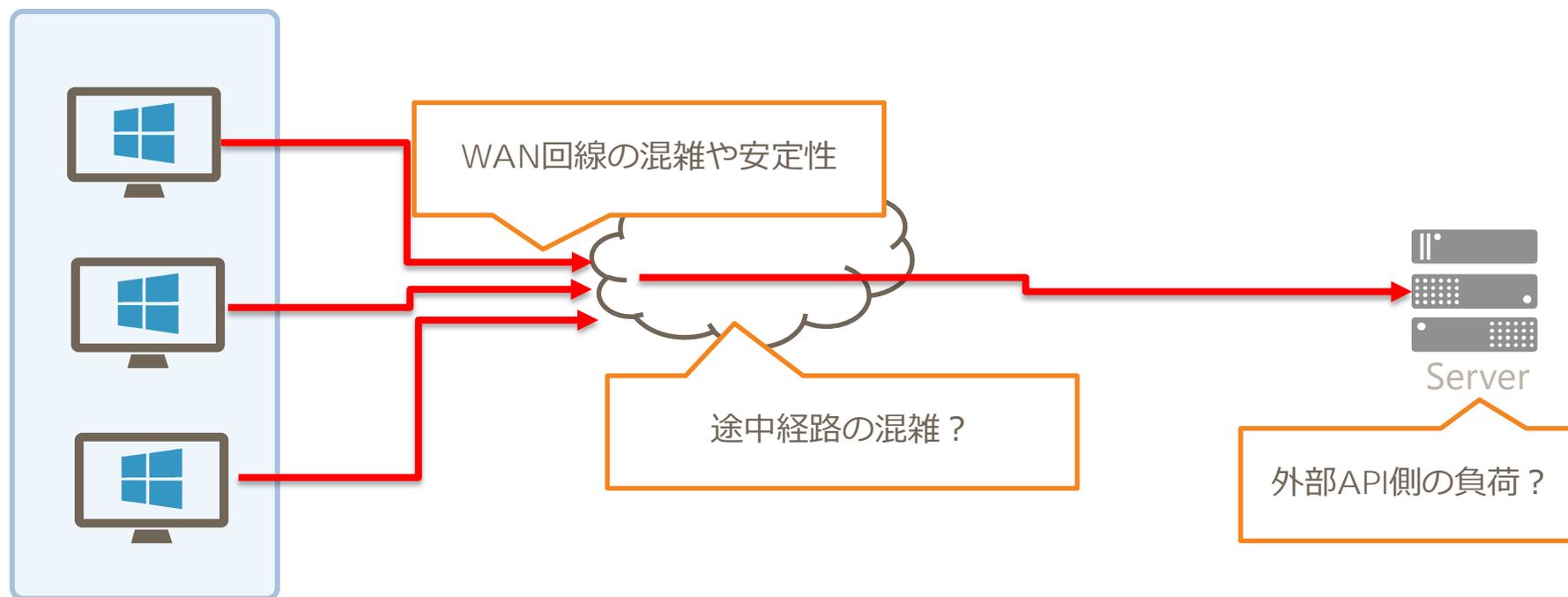
Path	Method	Status	Size	Time
https://api.twitter.com/1.1/search/tweets.json?...	GET	200	8.9kb	1s
https://api.twitter.com/1.1/search/tweets.json?...	GET	200	8.9kb	2s
https://api.twitter.com/1.1/search/tweets.json?...	GET	200	8.9kb	345ms

同じURLへのアクセスだが...

応答内容は同一だが
応答時間が大きく違う

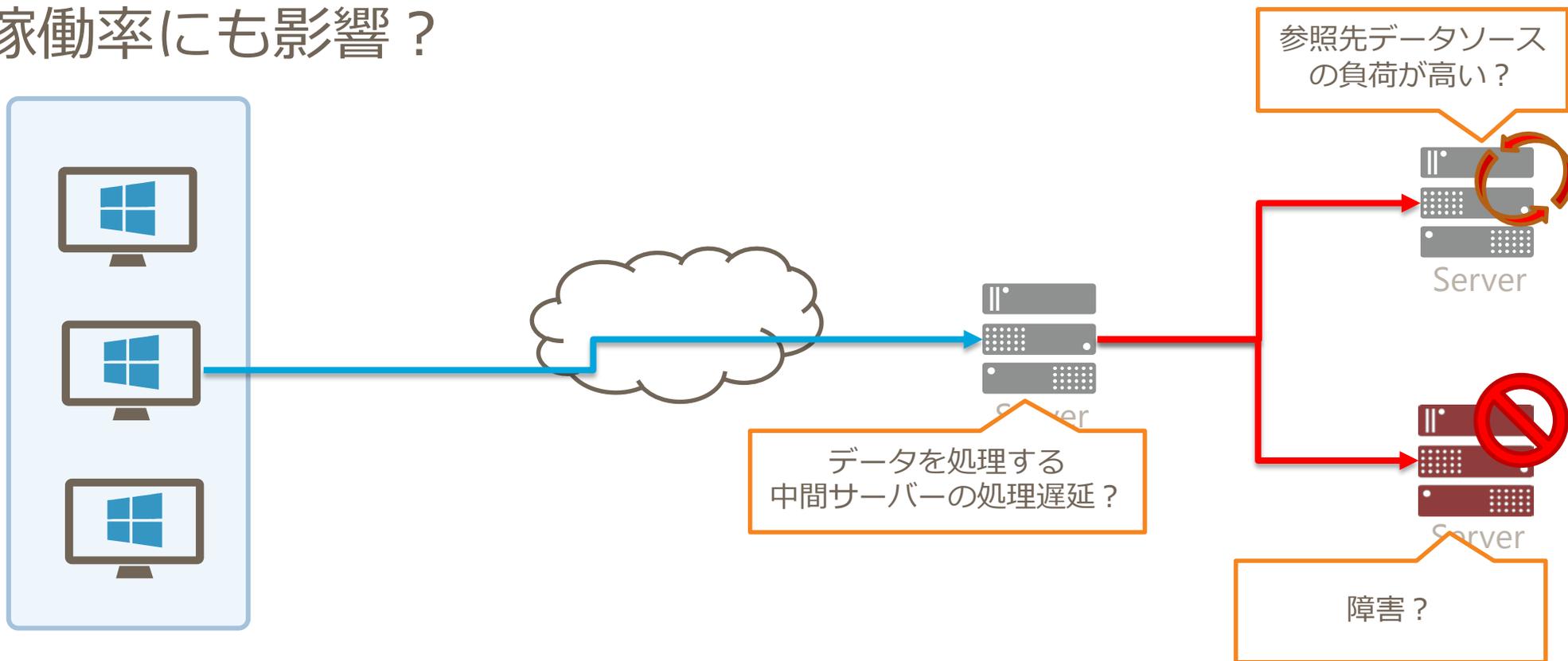
外部サービスのAPI利用を、より快適に – 課題

- 途中経路の安定性や混雑、外部API側の負荷などの要因で応答時間が変動する



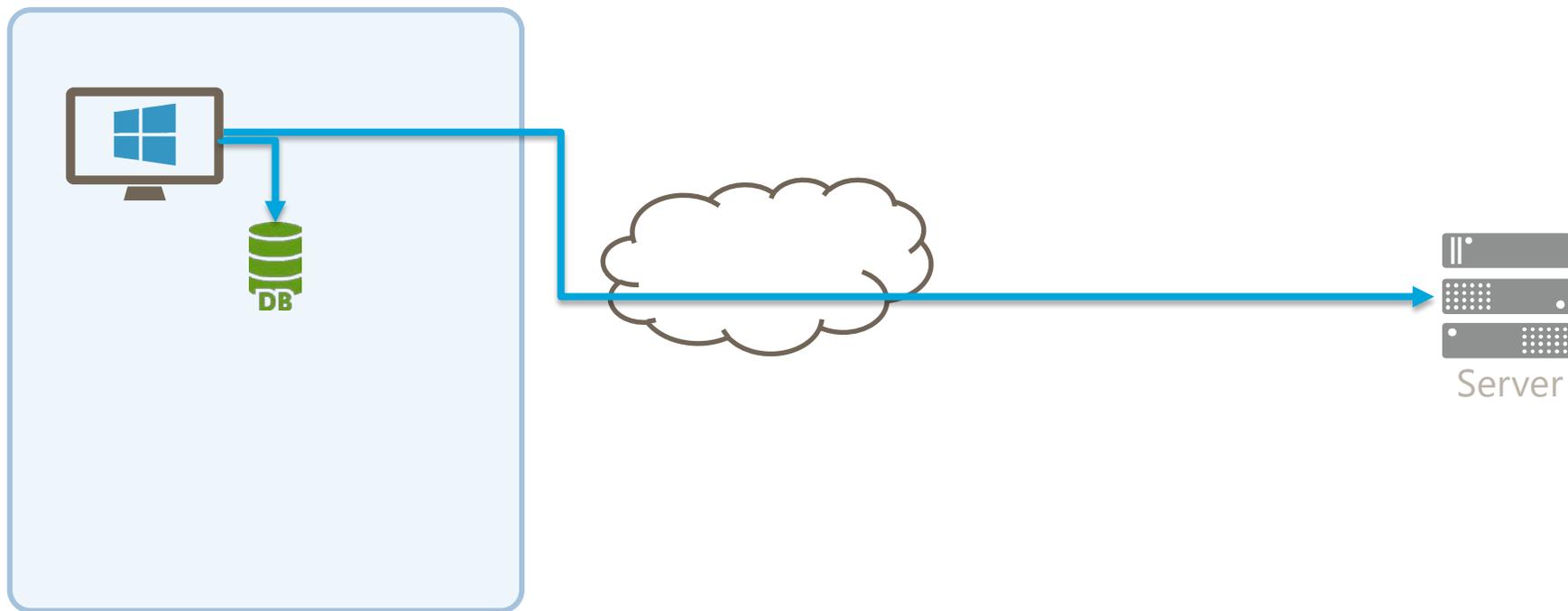
外部サービスのAPI利用を、より快適に – 課題

- 中間サーバーでデータを加工する場合は、中間サーバーの応答時間や稼働率にも影響？



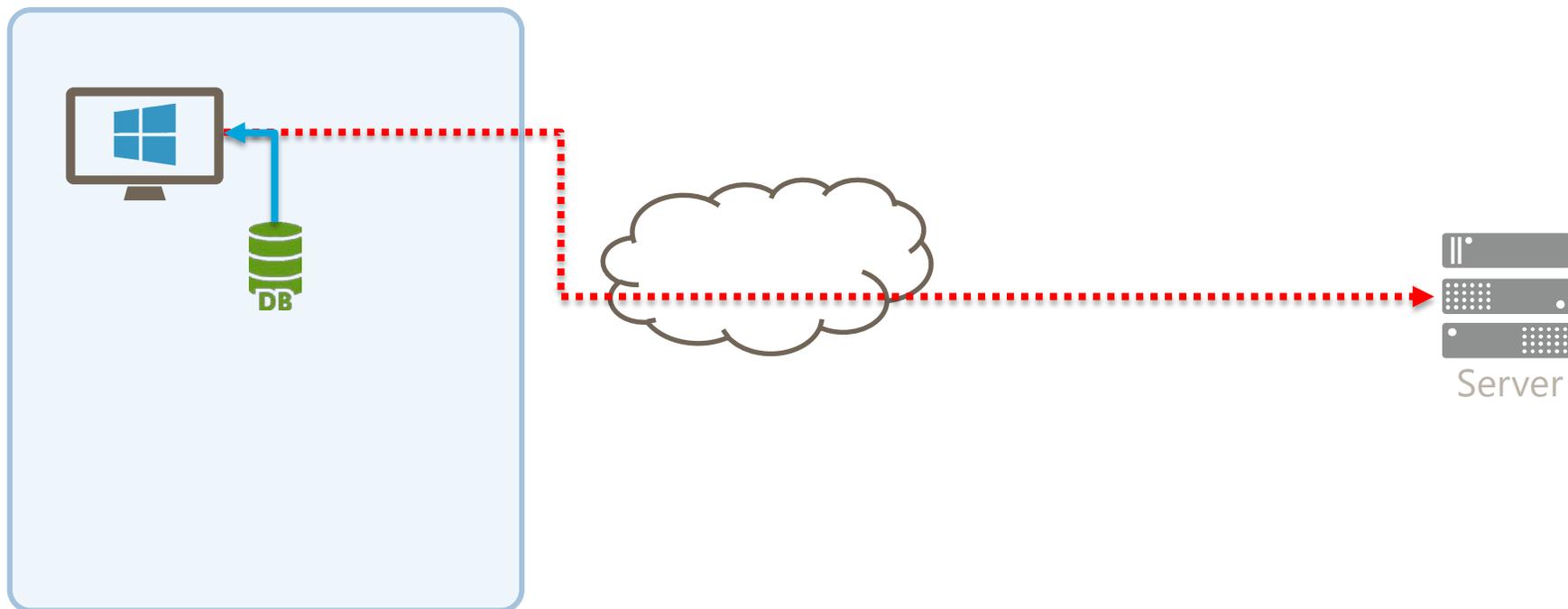
外部サービスのAPI利用を、より快適に – 解決策・提案

- 外部サービスから取得したデータをキャッシュできないだろうか？



外部サービスのAPI利用を、より快適に – 解決策・提案

- キャッシュがあれば、外部サービスとの通信が不安定な場合に代替可能



外部サービスのAPI利用を、より快適に – 選択肢

- 外部APIを利用するライブラリやコンポーネントにキャッシュ機能が無いのか？
 - キャッシュ機能がある場合は、要件に合致する機能かどうかを確かめる
- キャッシュの方法は.....
 - 永続性が必要ならデータベースを使う
 - 永続性が不要、データ量が少ないならオンメモリで (FDMemTable, FDLocalSQL)

■新たに作る業務アプリでスモールスタートからのスケーラビリティを考慮したい

- 新規に作るアプリで将来の需要が読みづらいものにデータベースサーバーを組み合わせるのは、構成が重すぎるかもしれません
- しかし将来のスケーラビリティを考えると、データベースサーバーでの利用は想定しておきたいところです



新たに作る業務アプリでスモールスタートからのスケールビリティを考慮したい - 課題

- 組み込みデータベースとデータベースサーバーではデータフォーマットが違う
 - SQLite のデータファイルは MS SQL, Oracle では当然使えない
- 運用ステージによって接続データベースを変更すると、改修が避けられない
 - データ自体の移行作業
 - アプリケーションの改修

新たに作る業務アプリでスモールスタートからのスケーラビリティを考慮したい – 解決策・提案

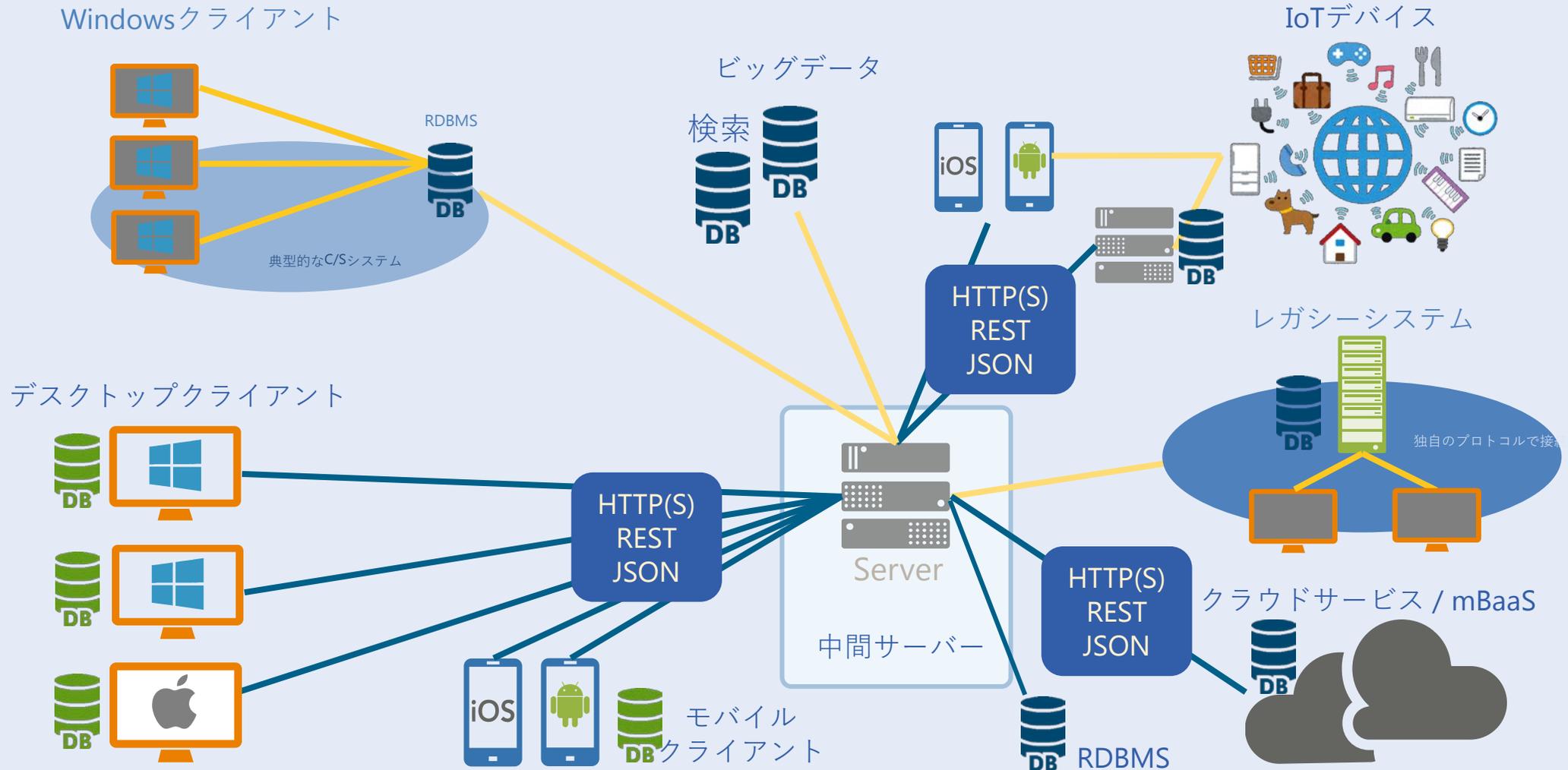
- 組み込み用とサーバー用で同じデータベースファイルが使える製品を選定する
 - InterBaseはIBLite、IBToGo、Desktop、Serverのいずれもデータベースファイルは共通フォーマット
- フェーズによって使い分けられる
 - 初期段階ではIBLite, IB ToGo, InterBase Desktop のいずれか
 - 規模が拡大したらInterBase Server に移行できる
 - 共通のファイルが使えるので移行の手間は少ない
- アプリケーションの改修も少ない
 - FireDACなら、FDConnectionの接続設定を変えるだけで基本の作業は終了

新たに作る業務アプリでスモールスタートからのスケーラビリティを考慮したい - 選択肢

- InterBase (IBLite, ToGo, Desktop, Server) なら組み込みからサーバーまでのニーズに対応できる

複合したビジネスモデルでのDB利用例

いたるところにデータベースが存在



まとめ

- これから作るアプリにはBDEではなくFireDAC
 - 遠隔の拠点では、ゼロアドミニストレーションDBでキャッシュやコピーの運用
 - モバイル向けでは、データ送信処理でのキャッシュ利用
 - 暗号化データベースを、業務アプリ向けデータ配布の手段として
 - 外部APIとの通信をキャッシュ利用
-
- データベースをサーバー・クライアントの枠組みではなく、「汎用的に使えるデータ置き場」や「暗号化が可能なデータ置き場」という観点で考えると、様々な用途に利用できる

THANKS!

www.embarcadero.com/jp

第34回 エンバカデロ・デベロッパーキャンプ