



Developer Camp

【C4】テクニカルセッション

「マルチデバイス向け
アプリ開発・構築における
データハンドリングのいろいろ」

エンバカデロ・テクノロジーズ
シニア・セールスコンサルタント 伊賀 敏樹

201411071011



はじめに

アジェンダ

- はじめに
- プログラムとデータ
- 企業データとの接続
- 中間サーバー
- より良いデータハンドリングのために
- まとめ

はじめに【1/3】

- スマートフォンやタブレットの普及
 - →マルチデバイス向けアプリ開発・構築はあたりまえの事に...
- プログラミングに関する点は忘れられずに注目されるが...
 - →データの永続化や取り回しなどのデータハンドリングが意外にも忘れられがち
画面は気にされるものの、データは忘れがちな傾向

はじめに【2/3】

- 情報システムは企業データに繋がる必要がある
 - → 企業データへの接続によって価値が生まれる
- 既存の他システムとの連携も重要
 - → HTTPS / HTTP 連携【中間サーバー】
 - SOAP / REST
 - → リレーショナルデータベース連携
 - Oracle、SQL Server、DB2、MySQL 等...
 - → ファイル連携
 - 既存システム、汎用機
 - その他...

はじめに【3/3】

- スマートフォンやタブレット等のモバイルアプリの特性
 - ネットワークが低速な状況で動作する必要
 - ネットワークが切断されている状況で動作する必要
 - モバイルアプリ側でデータを維持する必要
 - → 快適な操作性を実現するための工夫が必要
- これらの観点について一つずつ見ていく



プログラムとデータ

ネイティブと HTML

- 最近のモバイルアプリ・ゲームを観察していると
 - 基本的にネイティブ・アプリとして作成されている
 - 一部が素朴な HTML で実装されている
 - サーバ側で HTML を生成してクライアントで表示
 - 表示のみ、結果表示といった用途で利用されている
- モバイルアプリの更新がしづらい場合に、ネイティブと HTML を共存させる流行
 - **App Store** 等のアプリ更新頻度を向上できない諸々の事情が主因?
 - HTML により、あたかもプログラムとデータが一体化

アプリの更新しやすさ

- モバイルアプリは更新しづらい傾向にある
 - Web アプリ開発に慣れた人は要注意
- 「データ」について、プログラムにハードコードするのか、可変な項目として実装するのか、考慮が必要
 - 可変な項目だとして、都度通信して取得するのか、ローカルにキャッシュするのか?
 - 都度通信すると、結果として UX が大幅に悪化する懸念あり
- デスクトップアプリとの相違点のひとつ

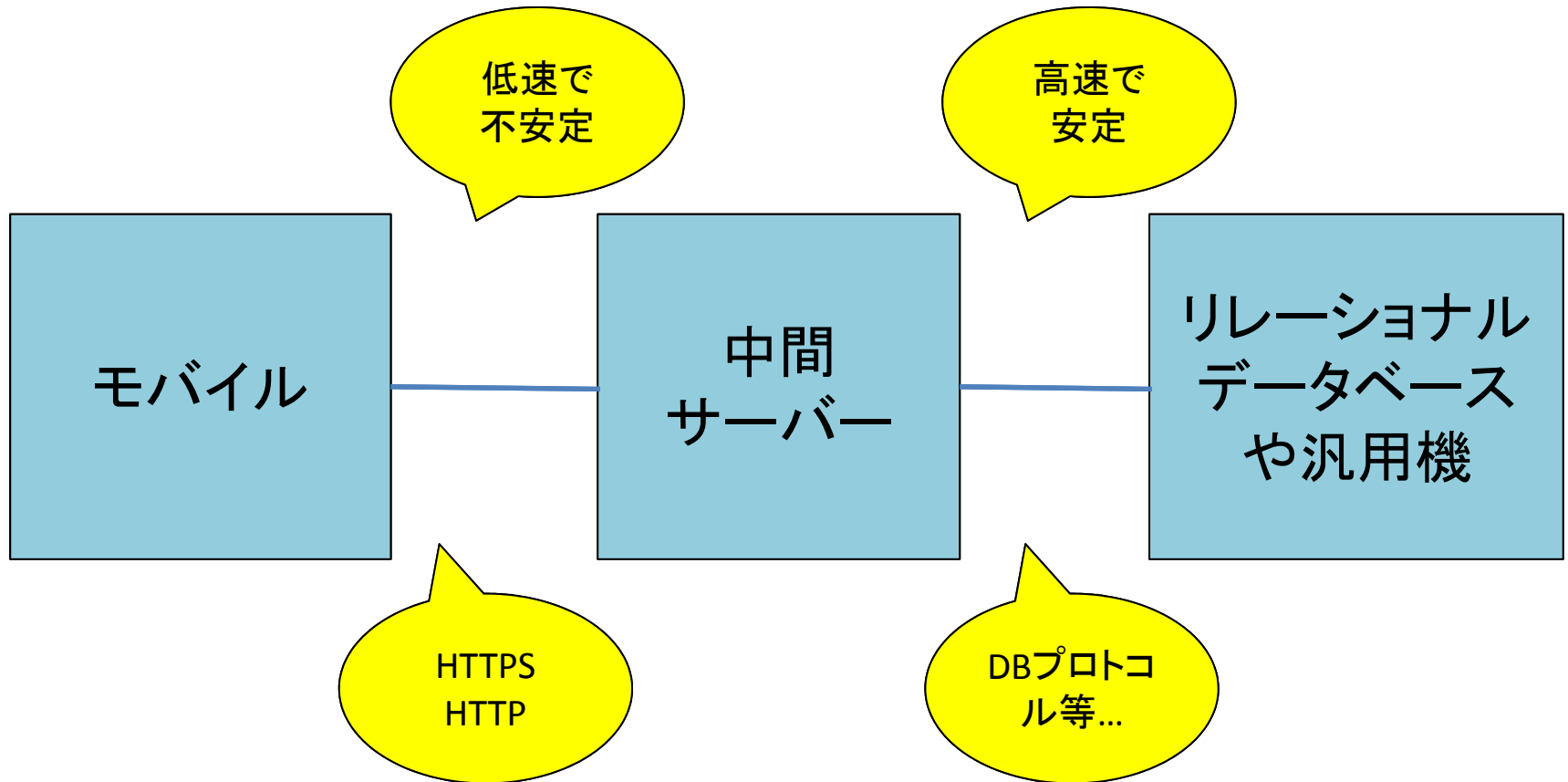


企業データとの接続

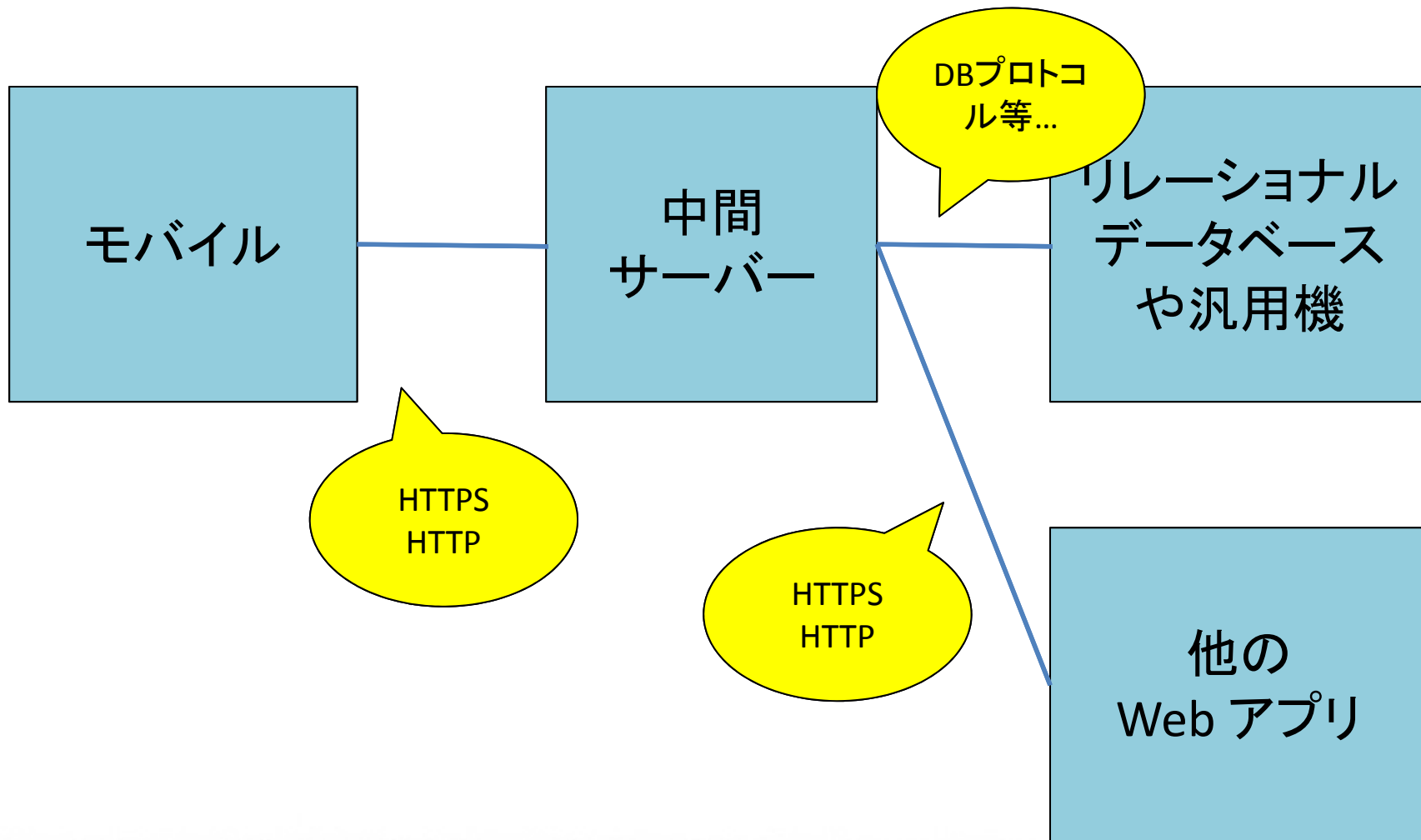
企業データとの接続【接続方式】

- モバイルアプリからの直接接続に“向いていない”と一般的に言われているもの
 - 汎用機
 - リレーショナルデータベース
 - ファイル
 - 他システムへの HTTPS / HTTP 直接接続
- 「通常」は自前の Web サーバー (HTTPS / HTTP 接続) 経由で間接的に...【中間サーバー方式】
 - → 新規構築の場合にコスト増に見えてしまう

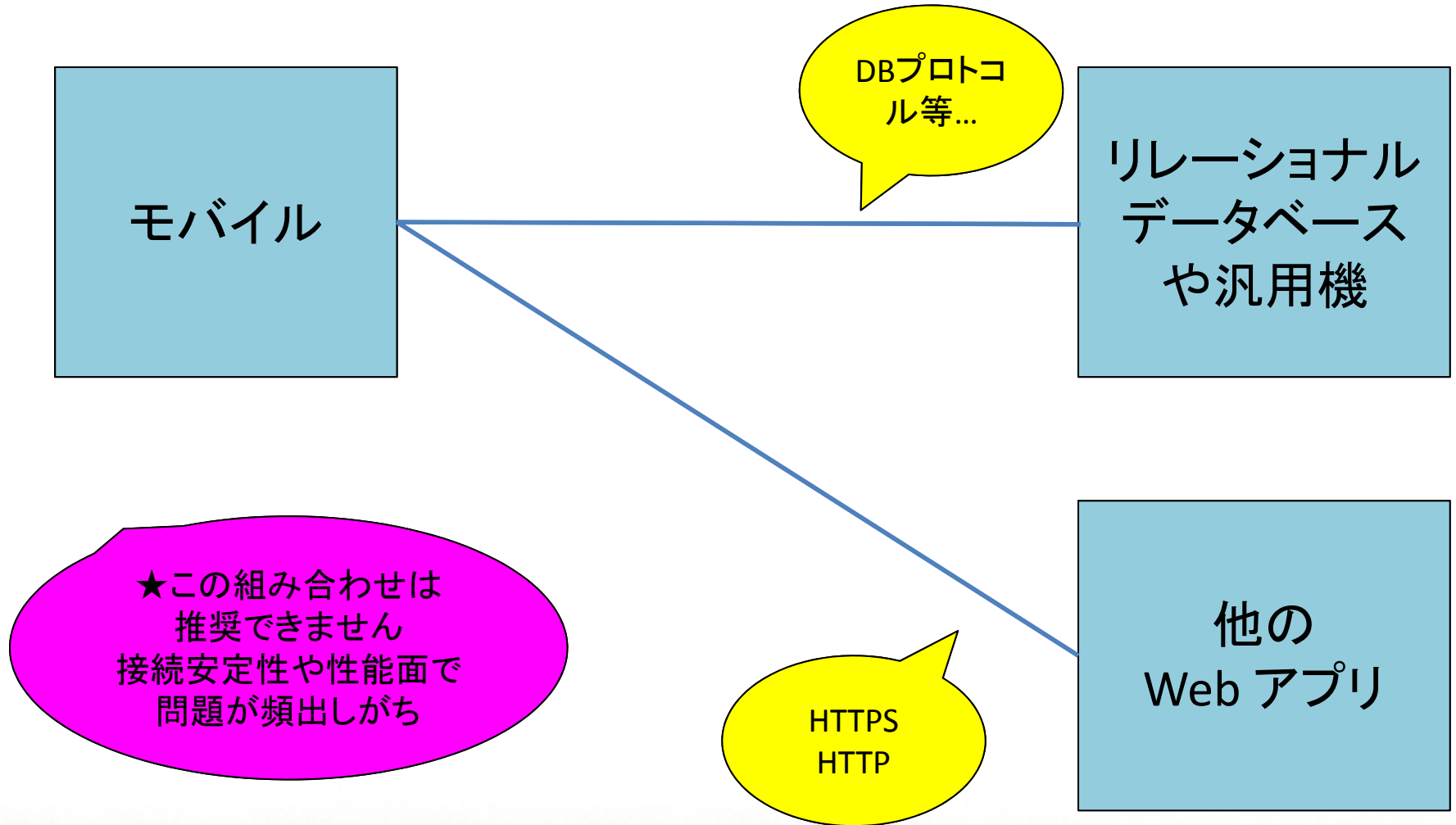
接続方式のイメージ【1/4】



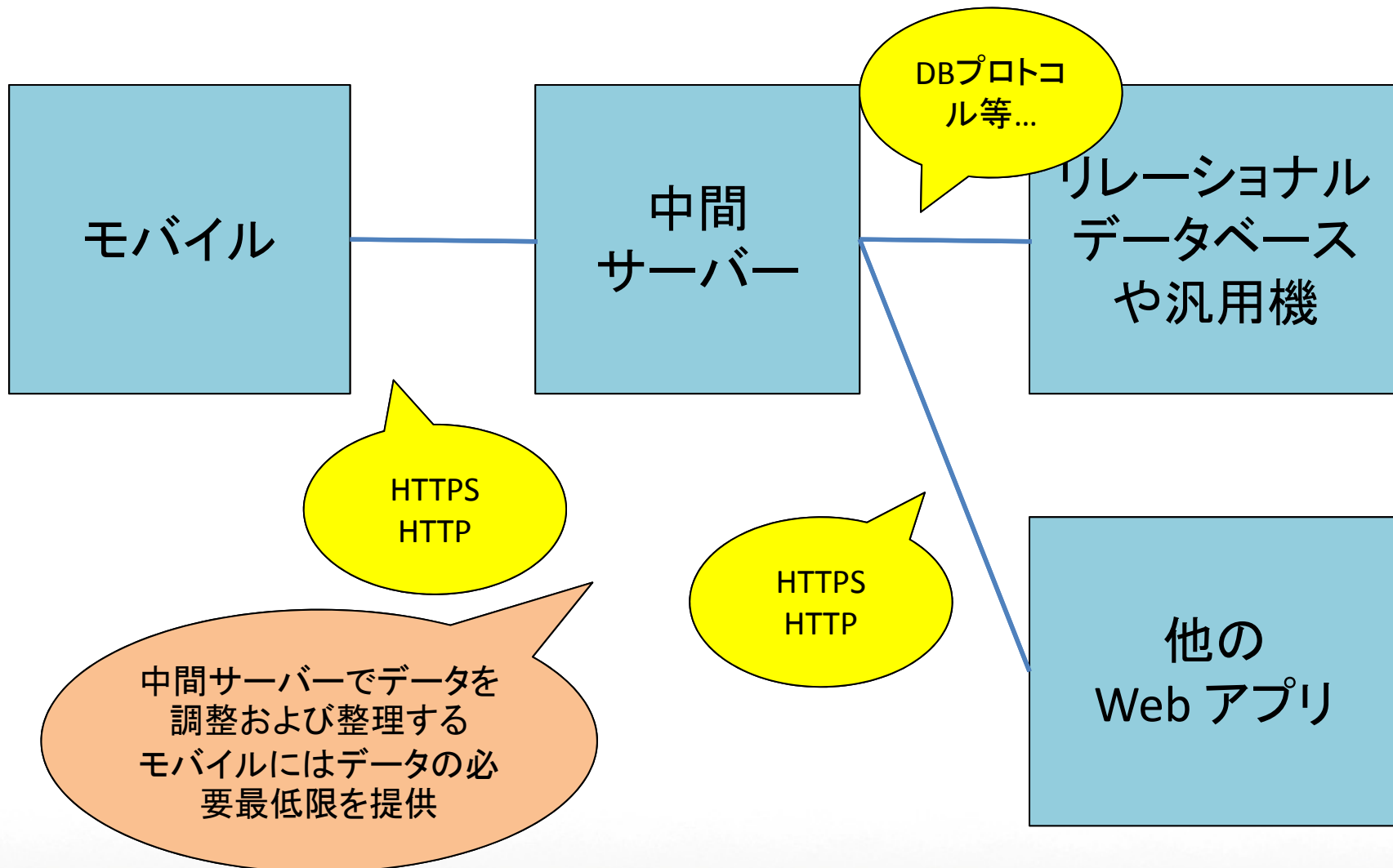
接続方式のイメージ【2/4】



接続方式の非推奨イメージ【3/4】



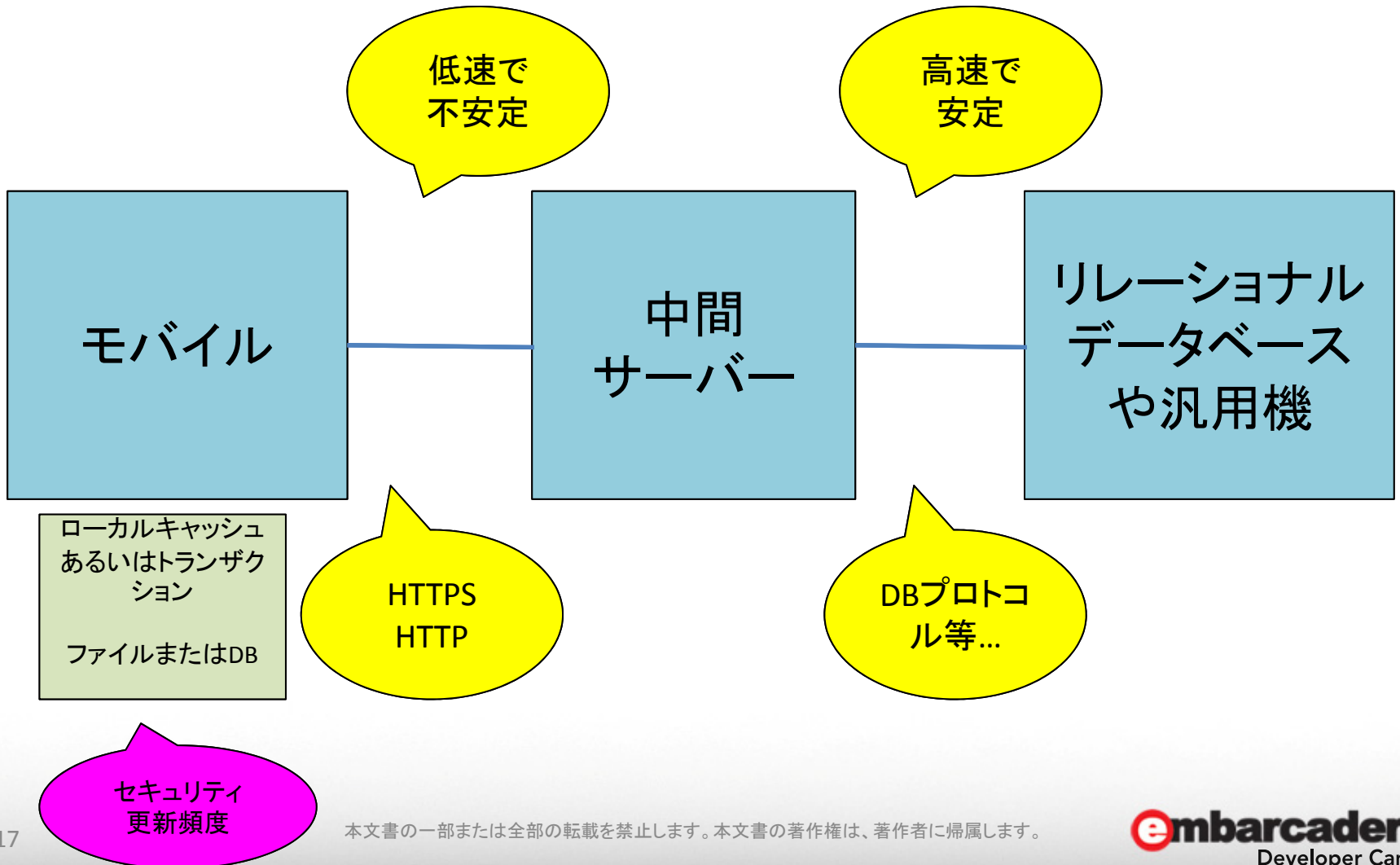
接続方式のイメージ【4/4】



データの特徴

- マスターデータおよびトランザクションデータ
 - 古典的な分類方法
- ローカルキャッシュおよびキュー
 - データの更新頻度および求められるデータ鮮度
- データの量 (大量には保持不可能)
- セキュリティ観点が必要かどうか
 - 暗号化
 - 削除の方法および契機
- そもそも、企業データとしての適切な取り扱い

データ保持



マスターデータ

- 企業データをモバイルアプリで受信
- “ある程度” の単位で受信
 - データを1件ずつ処理すると性能が出ない懸念
 - データを “ひとかたまり” で扱った場合に...
 - データの鮮度を判断する仕組み
 - データが壊れていないことを判断する仕組み
 - データのダイジェスト値の活用も効果が期待される
 - データ圧縮
- ネットワーク切断状態の考慮

トランザクションデータ

- モバイルアプリ側で発生したトランザクションデータの送信
 - “遅延化” できるかどうか
 - 順序性
 - 処理実施後の確認方法の提供
- 他のトランザクションデータの受信
 - データの鮮度 (マスターデータに類似する観点)
- ネットワーク切断状態の考慮

暗号化

- モバイルデバイスそのもののセキュリティ性
 - 例: リモートからの全削除
- 暗号化
 - ファイルの暗号化
 - データベースの暗号化

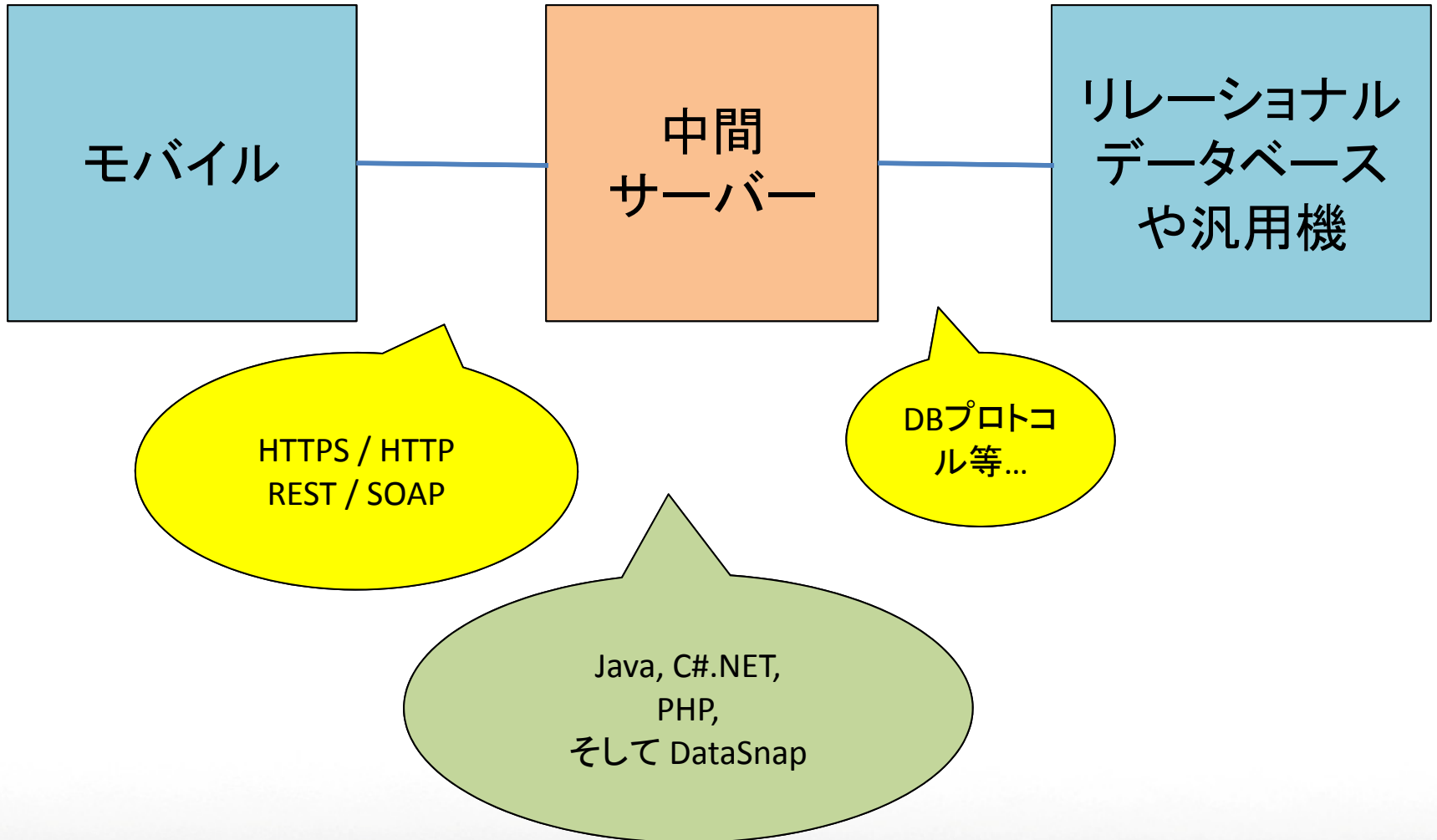


中間サーバー

中間サーバー

- 自前のリレーショナルデータベースに接続
- 他システムと接続
 - 汎用機
 - リレーショナルデータベース
 - ファイル
 - 他システムへの HTTPS / HTTP 直接接続
- HTTPS / HTTP による接続
 - SOAP / REST 形式
 - Java、.NET、そして PHP などの LT系
 - DataSnap 形式 (Delphi, C++Builder 系)

中間サーバー





より良い
データハンドリング
のために

より良いデータハンドリングのために

- Web アプリケーション型のソフトウェア開発とは異なることを理解する
- クライアント・サーバ型のソフトウェア開発とは異なることを理解する
- 必要なコストを予め適切に見積もる

- さまざまなロケーションに対するデータベース管理
 - ER/Studio Data Architect
- ローカルデータベース機能およびデータの暗号化
 - InterBase ToGo



まとめ

まとめ

- モバイルアプリは クライアントサーバー型とも Web アプリケーション型とも異なる
- 相違点を良く理解して、適切に設計しよう
- ツールなどにより手作りを回避できる部分を見極める



おわり