【セッションNo. 1】

基調講演

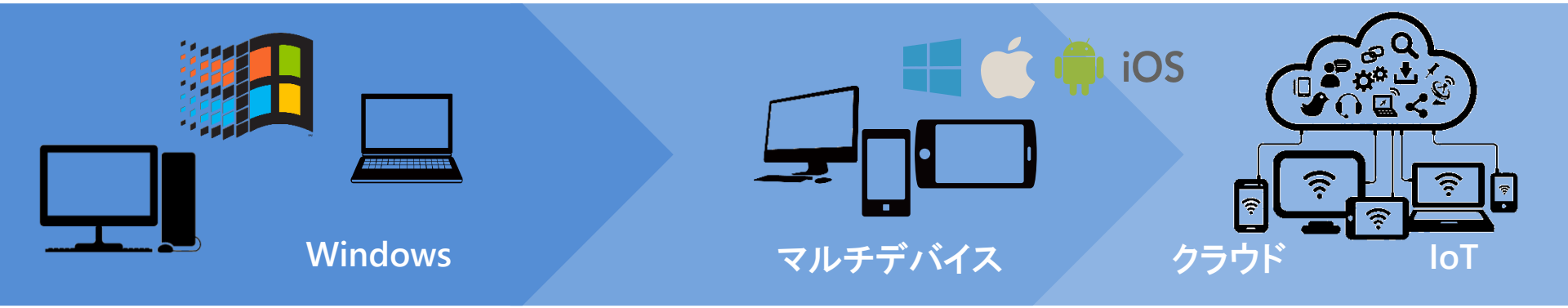
IoTでどう変わる？これからの業務システムとは

エンバカデロ・テクノロジーズ
日本法人代表
藤井　等 様

セールスコンサルタント
井之上　和弘 様

■ エンバカデロ・テクノロジーズについて

1995年、Windows向け開発ツールDelphiをリリース以来、
コンポーネントによるビジュアル開発によりソフトウェア開発を劇的に効率化



情報システムを取り巻く変化に対し、クラウドサポート、
マルチデバイスサポート、IoTサポートなどを強化してきました。

Delphiの進化

プラットフォームの
「進化」に適応

- Windows 10、OS X
- Android、iOS
- 最新のRDBMS

「接続性」の拡張

- ビッグデータ
- クラウド
- IoT

「多様化」への対応

- プラットフォームネイティブ
  コントロールの強化
- AppAnalytics
- Linuxサポート（サーバー側）

IT環境の進化、変化に対して、コンポーネント技術による
ビジュアル開発手法で効率化、単純化を推進

IoTとは


10
35


MINI JACK

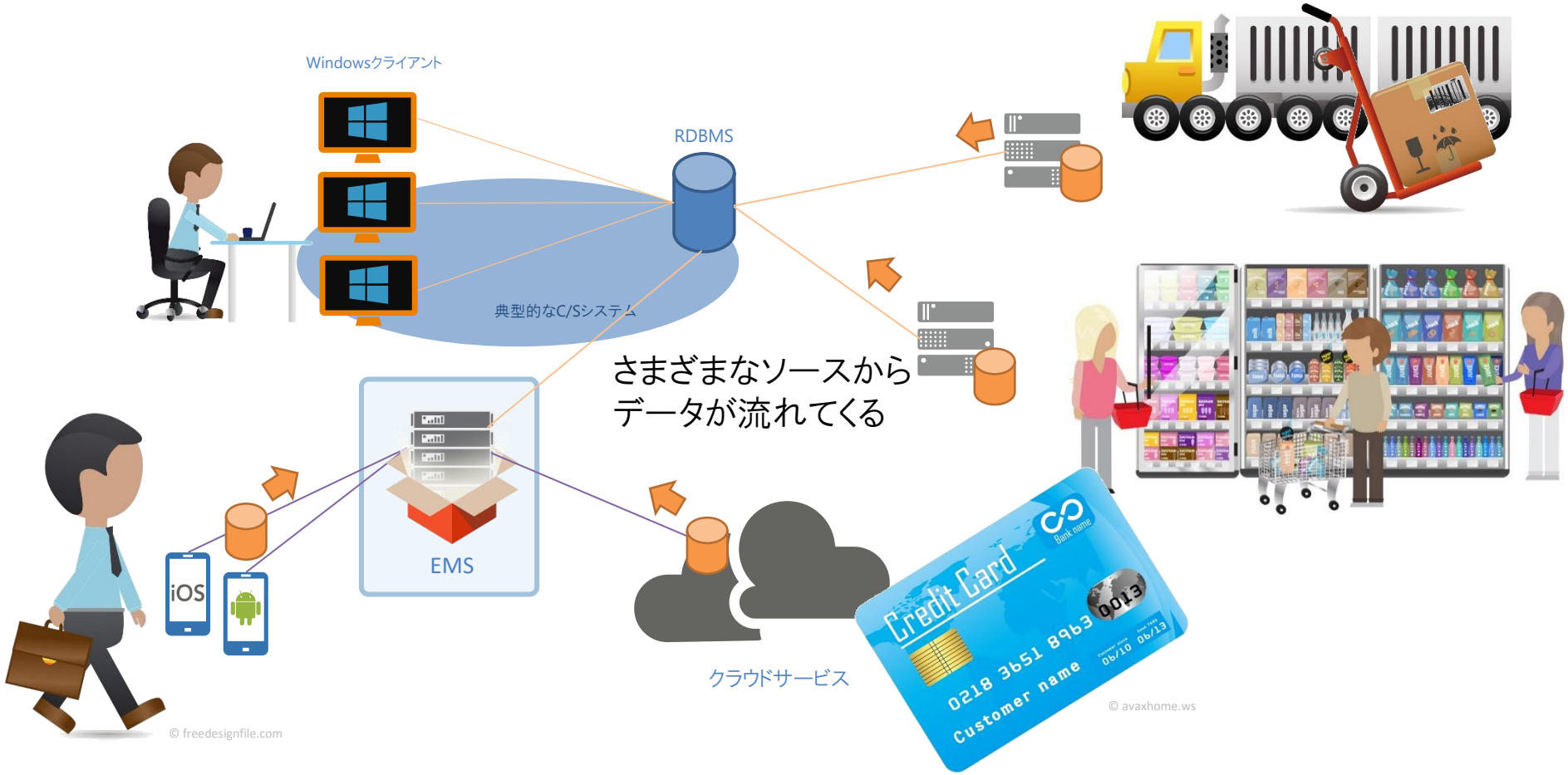

DATE
12-30
12:50
TIME
120
SYSTOLIC
mmHg
mmHg
80
DIASTOLIC
mmHg
M
48
72
PULSE
/MIN.
MODE
SET
START
/STOP
MEMORY


nest
HEATING
21

ソフトウェア開発者から見たIoT

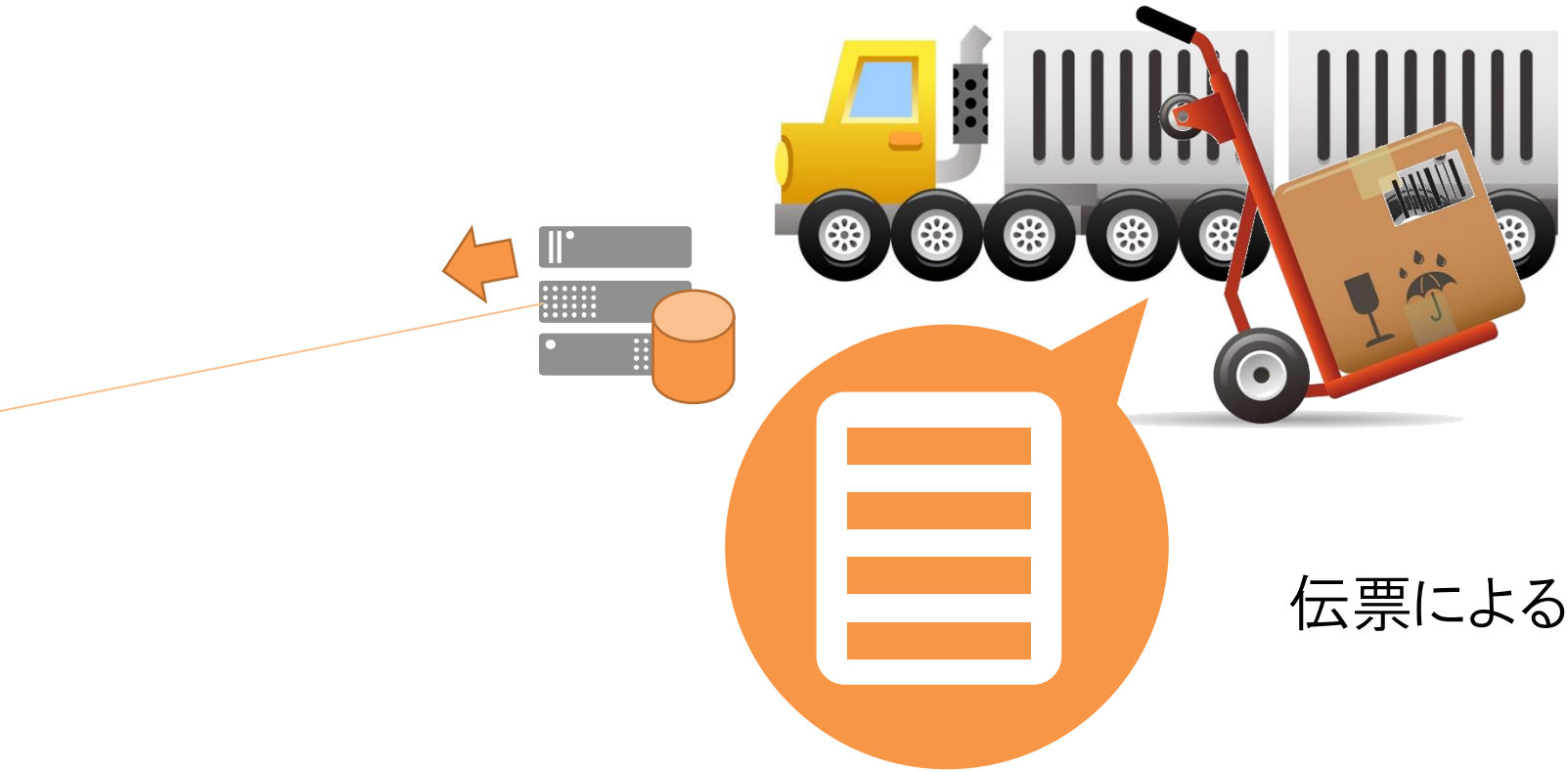
デバイスと通信できるAPIが提供されていると捉えることができる。

昨今のシステムは...


Windowsクライアント
RDBMS
典型的なC/Sシステム
さまざまなソースから
データが流れてくる
EMS
iOS
クラウドサービス
Credit Card
0218 3651 8963 0013
Customer name
© freedesignfile.com
© avaxhome.ws

従来の手法　1

紙ベースのデータを入力



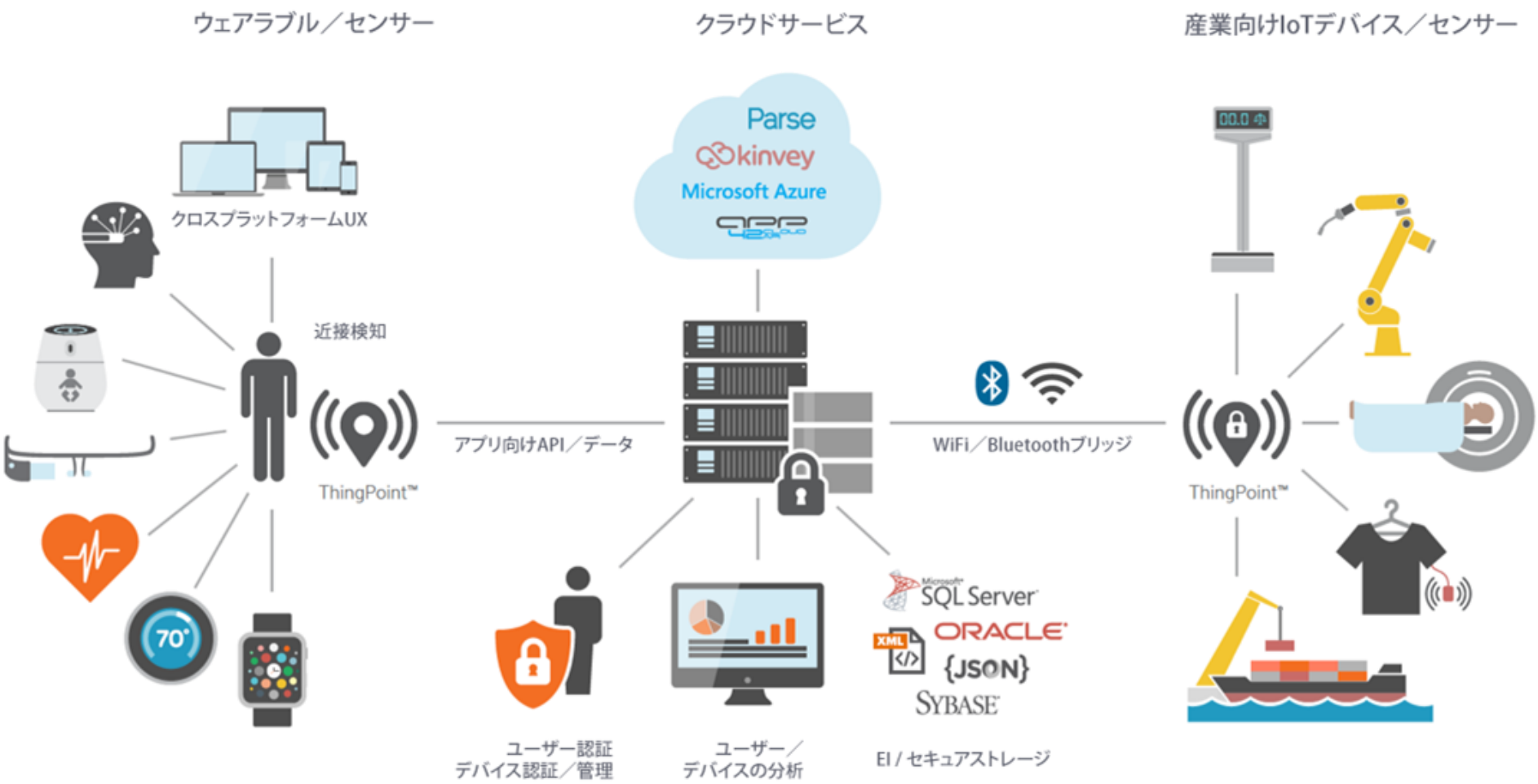
伝票による確認

従来の手法 2

IT化されているがデータの入力には人手が必要

手動操作による在庫確認

IoT時代のデータの流れは…



■ データ入力に関する比較

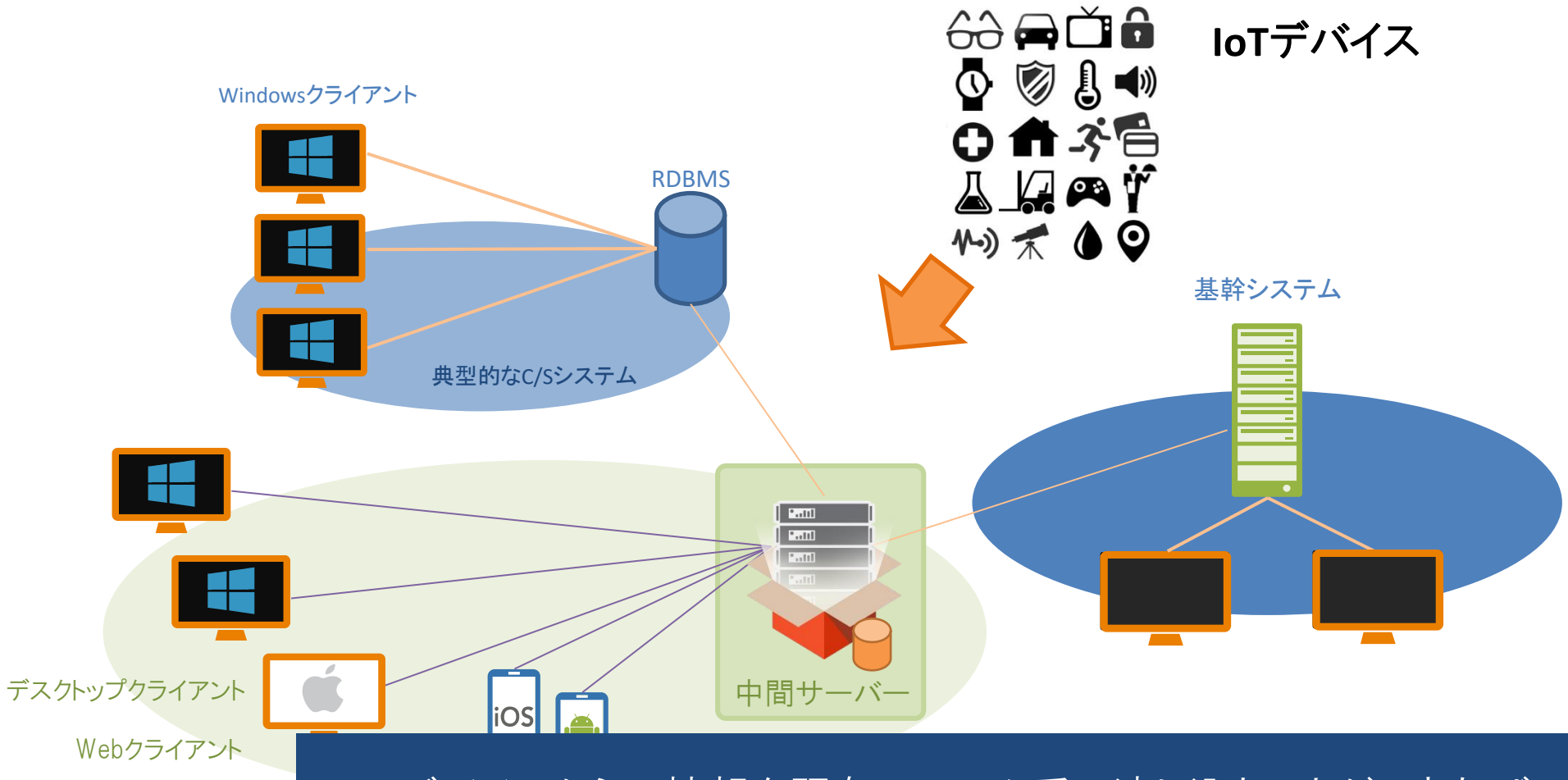
	従来手法	IoT
データの入力	人が介在して データを入力	情報を収集する デバイスから直接入力
入力のコスト	入力にかかる 人件費	ゼロ （デバイスのメンテナンスが必要）
リアルタイム性	データ入力が完了する までのタイムラグ	リアルタイム
フィードバック	タイムラグがあるため 限定的な効果	デバイスを直接制御し 即応可能
データの粒度	必要な情報のみ 入力	すべての情報を 入力
フィルタリング	人が介在することで 重要性を判断	ソフトウェアによる フィルタリングが必要

■ 業務における入力されなかったデータ

- 従来は入力コストやリアルタイム性が実現できないなどの問題から入力することを検討すらされなかったデータ
  - 工場内での従業員の行動
  - 在庫品の状況、移動
  - 在庫量の変化
  - 温度変化
  - 顧客の移動、滞留
    :

IoTを活用することで、これまで入力されなかったデータを取得し、業務をより効率化、最適化することができるかもしれない！

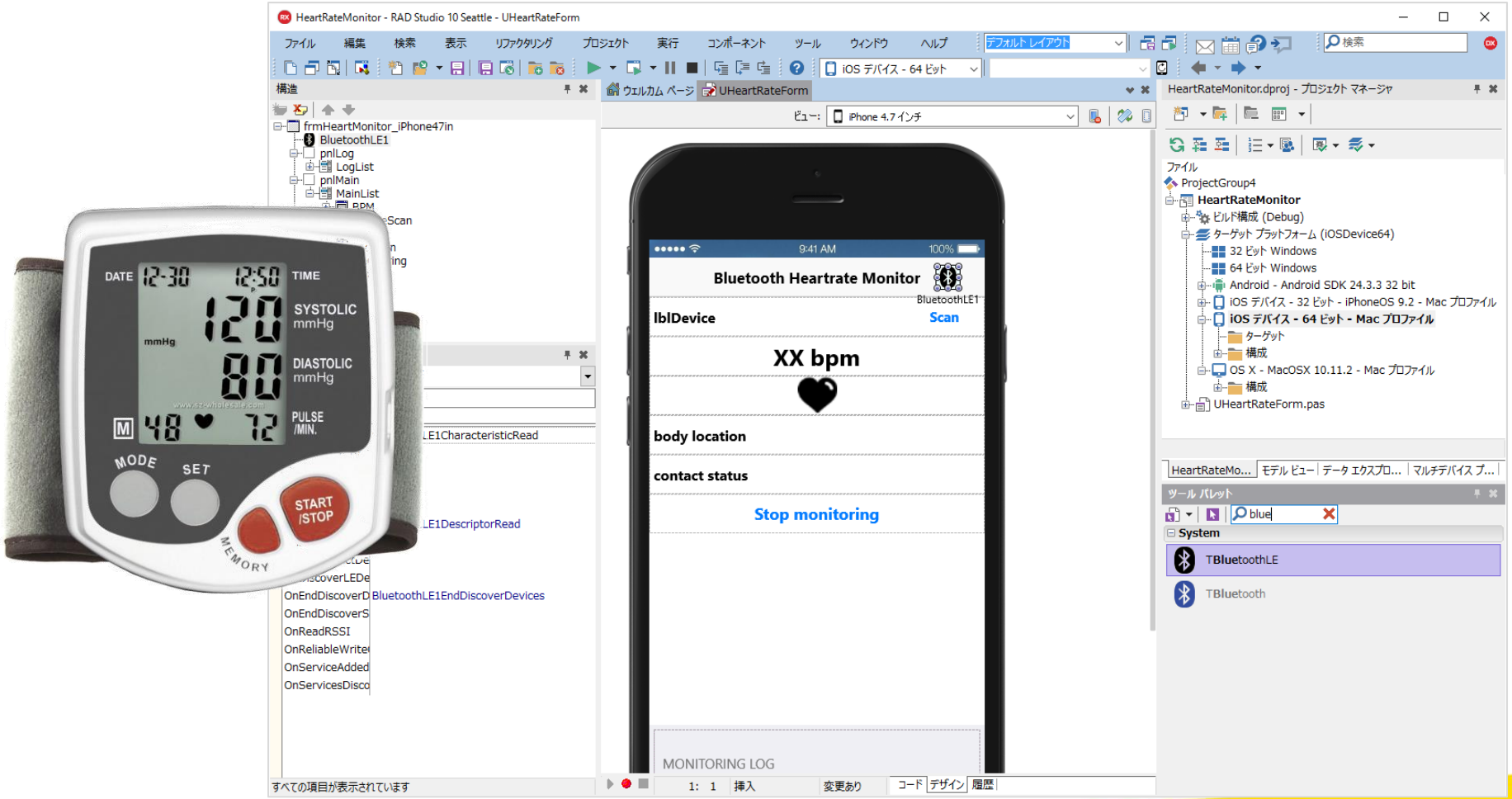
SFを現実に引き寄せてみましょう



IoTデバイスからの情報を既存システム系に流し込むことができれば
IoTのメリットを活かしたシステムの構築が可能になります。

IoTデバイスを利用する具体的な方法

- Delphi / RAD Studioならコンポーネントがあります！



IoTデバイスの基礎知識

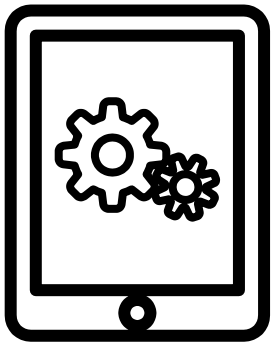
IoTデバイスでできること

- 状態を把握する
  - 温度・湿度 、加速度、傾きなどのセンサーの値
  - On/Off などの状態
  - 消費電力などの機器のステータス
  - フィットネス、ヘルスケア系のセンサーの値取

- 距離・位置を把握する
  - デバイスが発した電波強度と受信強度の差から距離を推定3箇所以上の固定ビーコンとの距離から、位置を算出（ビーコンと受信機側が逆でも同様）
  - 物品管理（ビーコンを設置した機器がどこにあるか）、屋内ナビゲーション（いまどこにいるか）、行動履歴収集（どのように移動したか）などに利用可能

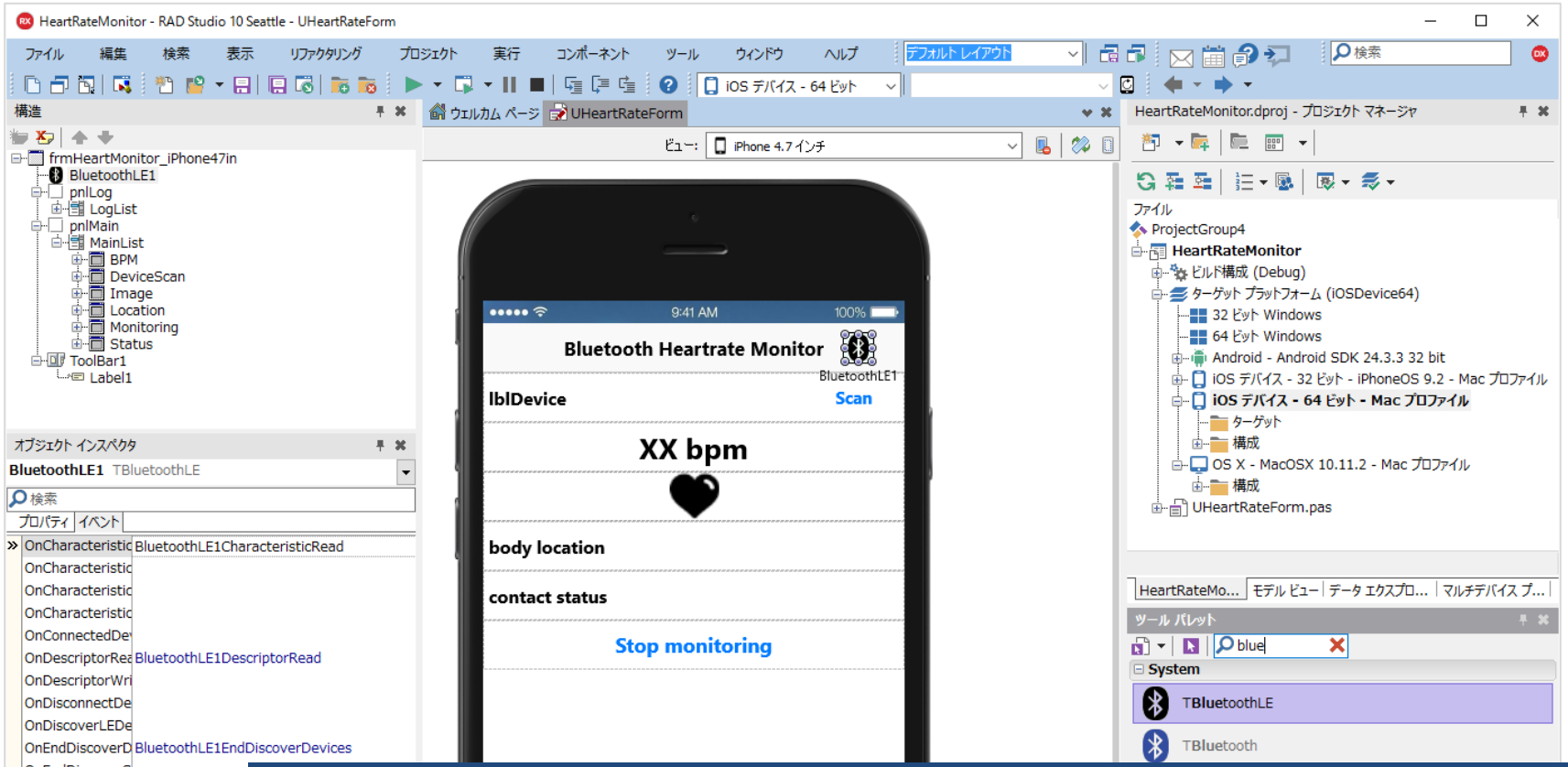
IoTデバイスと従来のセンサーの違い

- 設置に関する自由度が高い
  - 物理的な配線が不要
  - 設置数を増やすことが容易
  - 設置位置の移動が容易
  - ただし位置を移動した場合は、そのデータを使用するアプリケーション側の情報も必要に応じて更新する
- 受信機にBLE対応デバイスが利用できる
  - PCやスマホでデータを読み取り可能
  - 個々の目的に合わせたアプリケーションでデータを収集できる



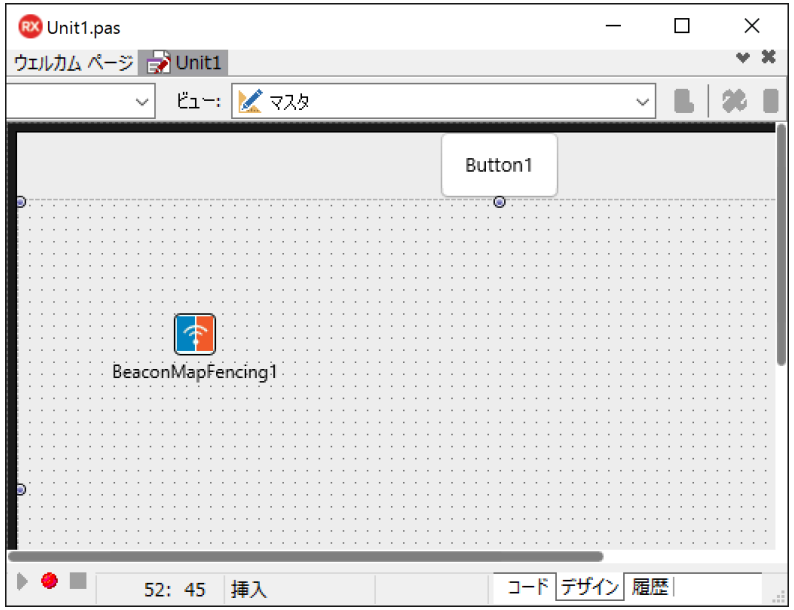
IoTアプリケーションの開発

IoTデバイスとの通信



Delphiに用意されたBluetoothおよびBluetooth LEコンポーネントを用いれば、デバイスとの通信を容易に実装できます。

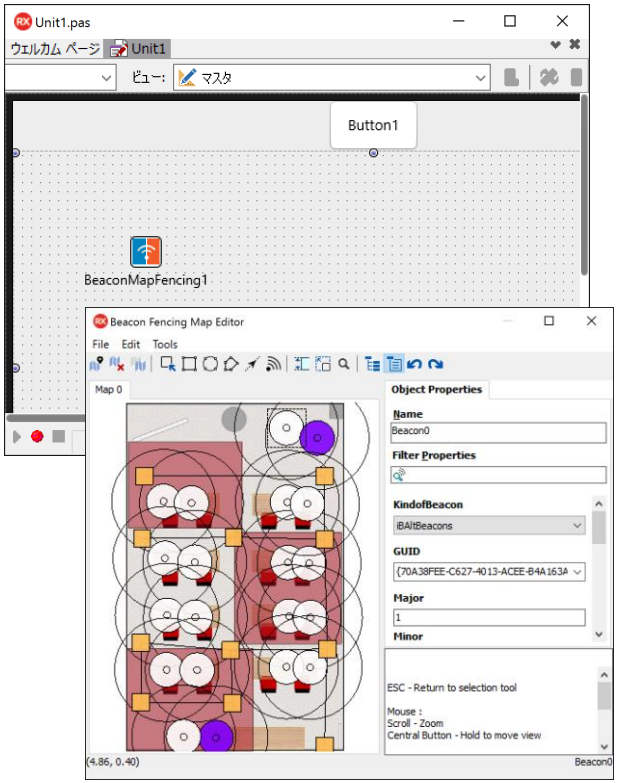
■ より専門性の高いIoTデバイスの利用



- ビーコンの利用
  - TBeaconコンポーネントによりビーコンとの通信を容易に実装可能
- ビーコンによる位置検出
  - BeaconFenceコンポーネントを用いれば、複数のビーコンを用いて、移動するスマートフォンの現在位置を検出可能

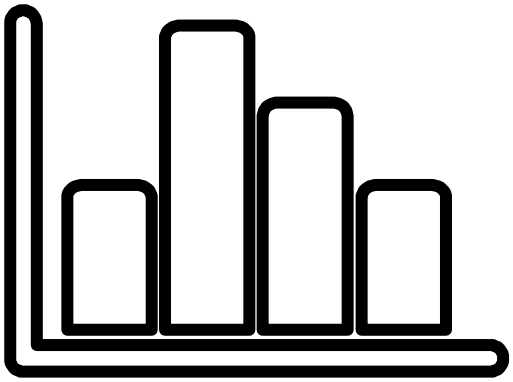
ビーコンに特化したコンポーネントを用いることで、
ビーコンを活用したアプリケーションを素早く開発可能

■ ビーコンによる位置検出（BeaconFence）



- 通常の測位アプリ作成手順：
  - ビーコンの固有情報と設置位置データを作成
  - 検出したビーコンとの受信側との距離を計算
  - ビーコンの位置と距離から三辺測量より受信側の位置を推定
- ビーコンを仮設置して試験し、検出精度の問題が出た場合はビーコンの設置場所を変更してアプリを再度試験

RAD Studio / Delphi + BeaconFence を用いれば、フロアレイアウトに対してビーコンを配置するだけで基本的な実装が完了



IoTデータの活用と注意点

■ 保存されたデータの活用

- 履歴
  - さまざまなデータを時系列順に記録しておく
  - 特定の時間帯のさまざまなデータを横断して、過去に発生した事象の調査を行うことで、潜在的な問題点が見えてくることも

	時系列順の変化	測定値																				平均値	中央値	標準偏差
温度 ℃		25.0	24.7	24.2	23.5	23.7	23.5	22.9	22.3	21.6	21.1		19.7	19.4	20.2	20.6	21.0	20.5	20.5	19.8	20.2	20.2	21.5	21.0	1.74
湿度 %		40.0	39.7	39.3	39.3	39.7	38.9	38.6	39.3	39.7	39.4		40.8	41.4	40.7	41.4	42.1	41.4	40.6	40.0	40.7	40.0	40.1	40.0	0.93
座標: x		30.0	29.5	28.7	28.3	27.7	28.6	28.0	28.7	28.0	28.8		27.2	26.6	27.0	26.6	26.0	26.9	26.4	26.4	27.0	28.0	27.8	28.0	1.02
座標: y		30.0	29.4	28.9	28.8	29.6	29.7	29.7	30.1	29.1	28.3		29.2	29.1	30.0	29.9	30.8	30.6	29.6	29.0	28.0	28.3	29.3	29.2	0.73
加速度: x		0.0	0.2	0.4	0.3	0.0	0.0	0.4	0.4	0.8	1.2	.....	2.2	2.1	2.4	2.6	2.7	2.5	2.3	2.8	3.2	3.4	1.5	1.7	1.10
加速度: y		0.0	-0.2	-0.3	-0.5	-0.5	-0.7	-0.5	-0.5	-0.6	-0.3		-1.3	-1.6	-1.5	-1.2	-1.7	-1.5	-1.6	-1.3	-1.8	-1.3	-0.9	-0.6	0.57
加速度: z		0.0	0.2	-0.3	-0.5	-0.4	-0.7	-1.1	-1.2	-1.2	-1.4		-1.5	-1.0	-1.5	-1.4	-1.3	-0.8	-0.5	-0.1	0.2	0.1	-0.8	-1.1	0.58
明るさ lux		750.0	750.7	749.9	750.8	751.0	751.9	751.8	752.3	753.3	753.3		754.2	753.7	752.7	753.5	753.5	753.4	753.5	752.9	753.4	753.4	752.7	753.4	1.28
消費電力 W		1000.0	1000.2	1000.7	999.9	1000.4	1001.2	1001.9	1001.5	1000.7	1001.4		1003.0	1002.3	1002.3	1001.9	1002.5	1003.2	1003.8	1004.6	1003.6	1004.6	1002.0	1001.9	1.32

■ 保存されたデータの活用

- 異常検知
  - 過去の統計データとの違いから、現時点の状態の正常･異常を判定する

	時系列順の変化		過去実績の指標																				直近の指標		
	過去実績値	直近の変化	平均値	中央値	標準偏差	測定値																	平均値	中央値	標準偏差
温度 °C			21.5	21.0	1.74	19.7	19.4	20.2	20.6	21.0	20.5	20.5	19.8	20.2	20.2		26.2	26.9	27.3	27.3	28.1	29.0	29.9	30.0	30.9	31.8	28.8	28.6	1.88
湿度 %			40.1	40.0	0.93	40.8	41.4	40.7	41.4	42.1	41.4	40.6	40.0	40.7	40.0		43.5	43.8	43.5	44.2	44.8	45.0	44.4	44.1	44.1	44.0	44.1	44.1	0.50
座標: x			27.8	28.0	1.02	27.2	26.6	27.0	26.6	26.0	26.9	26.4	26.4	27.0	28.0		29.6	30.1	30.4	31.1	30.2	30.7	29.9	28.9	29.2	28.9	29.9	30.0	0.74
座標: y			29.3	29.2	0.73	29.2	29.1	30.0	29.9	30.8	30.6	29.6	29.0	28.0	28.3		33.0	32.2	32.0	32.0	32.7	32.8	32.7	33.0	32.8	33.6	32.7	32.8	0.51
加速度: x			1.5	1.7	1.10	2.2	2.1	2.4	2.6	2.7	2.5	2.3	2.8	3.2	3.4	.....	1.5	1.1	1.5	1.3	0.8	0.6	0.1	0.2	0.2	0.3	0.8	0.7	0.56
加速度: y			-0.9	-0.6	0.57	-1.3	-1.6	-1.5	-1.2	-1.7	-1.5	-1.6	-1.3	-1.8	-1.3		2.2	2.4	2.6	2.6	3.0	2.8	2.9	3.3	3.6	3.1	2.8	2.9	0.42
加速度: z			-0.8	-1.1	0.58	-1.5	-1.0	-1.5	-1.4	-1.3	-0.8	-0.5	-0.1	0.2	0.1		1.7	1.7	1.3	1.3	1.1	1.2	1.1	0.9	1.2	1.1	1.2	1.2	0.25
明るさ lux			752.7	753.4	1.28	754.2	753.7	752.7	753.5	753.5	753.4	753.5	752.9	753.4	753.4		748.7	747.8	747.3	747.9	747.4	747.3	747.2	747.8	747.0	747.8	747.6	747.6	0.49
消費電力 W			1002.0	1001.9	1.32	1003.0	1002.3	1002.3	1001.9	1002.5	1003.2	1003.8	1004.6	1003.6	1004.6		997.3	997.5	997.9	998.4	998.7	999.2	999.2	999.6	1000.4	1000.5	998.9	998.9	1.11

■ 保存されたデータの活用

- 将来予測
  - 過去データの履歴から、将来の予測を行う
  - 予測用の専用のアルゴリズムを実装する、AIや機械学習を利用する、などの方法が考えられる

	時系列順の変化	過去実績の指標																						
	過去実績値	平均値	中央値	標準偏差	測定値																			
温度 °C		21.5	21.0	1.74	24.8	25.2	26.0	26.0	25.9	25.8	25.6	25.1	24.9	25.3		26.2	26.9	27.3	27.3	28.1	29.0	29.9	30.0	30.9	31.8
湿度 %		40.1	40.0	0.93	45.8	44.8	44.5	44.7	45.0	44.5	44.2	43.6	43.2	43.2		43.5	43.8	43.5	44.2	44.8	45.0	44.4	44.1	44.1	44.0
座標: x		27.8	28.0	1.02	30.7	31.0	30.3	30.2	30.1	29.7	29.0	29.6	28.9	29.1		29.6	30.1	30.4	31.1	30.2	30.7	29.9	28.9	29.2	28.9
座標: y		29.3	29.2	0.73	30.1	29.7	30.6	30.9	31.7	31.6	31.5	32.4	33.1	32.9		33.0	32.2	32.0	32.0	32.7	32.8	32.7	33.0	32.8	33.6
加速度: x		1.5	1.7	1.10	0.5	0.7	0.7	0.5	0.6	1.0	1.1	1.1	0.8	1.2	.....	1.5	1.1	1.5	1.3	0.8	0.6	0.1	0.2	0.2	0.3
加速度: y		-0.9	-0.6	0.57	1.5	1.7	1.8	1.5	1.4	0.9	1.0	1.3	1.6	1.3		2.2	2.4	2.6	2.6	3.0	2.8	2.9	3.3	3.6	3.1
加速度: z		-0.8	-1.1	0.58	1.2	1.5	1.8	1.8	1.7	1.4	1.8	1.4	1.2	1.2		1.7	1.7	1.3	1.3	1.1	1.2	1.1	0.9	1.2	1.1
明るさ lux		752.7	753.4	1.28	750.4	749.9	749.3	749.7	750.1	749.6	748.9	747.9	748.7	748.3		748.7	747.8	747.3	747.9	747.4	747.3	747.2	747.8	747.0	747.8
消費電力 W		1002.0	1001.9	1.32	997.6	997.4	997.3	997.6	998.6	997.7	997.1	997.9	997.4	996.7		997.3	997.5	997.9	998.4	998.7	999.2	999.2	999.6	1000.4	1000.5

保存されたデータの活用

- 過去データの履歴を蓄積しておけば、今後新たな手法が登場した場合に、異なる切口で分析可能
- ただしIoTデバイスの投入数が増えるほどログのデータ量も増えるため、このことが問題となる場合はデータの間引きを検討
- プライバシー（位置、速度、経路、その他ヘルスケア関連情報など）に紐づく可能性のある情報の取り扱いに留意

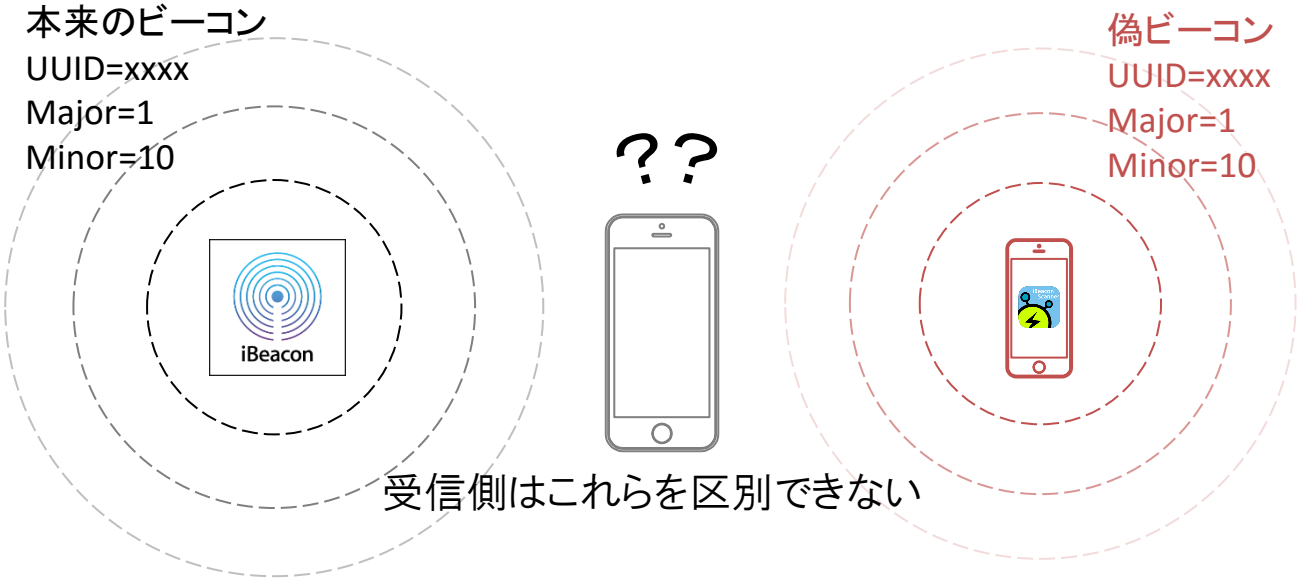
IoTデバイスの注意点

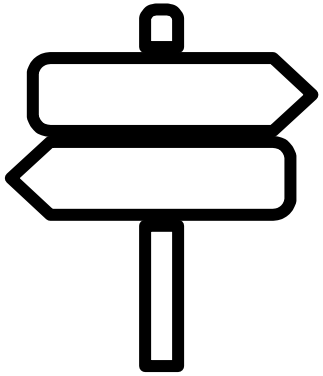
- データ取得の際に認証がない
  - BLEでは事前のペアリングは必須ではない（従来のBluetoothはペアリングが必須）
  - あらかじめ認証されたデバイスだけに限って情報提供することはできない
  - 万が一、関係者以外が取得しても差支えのない情報のモニタリングに用いる。（ただし、IoTデバイスで取得できるデータは単なる値にすぎず、他の情報との組み合わせることに意味があるので、第三者による値の取得を極度に恐れる必要はない）

- IoTデバイスが発した情報が、すべて取得できることを保証できない
  - 電波を用いる以上、電波が受けられない状況が発生しうる
  - BLEデバイスが使用する周波数帯は2.4GHzなので、同じ周波数帯を使うデバイス（電子レンジ、WiFiなど）との干渉が起こりうる
  - ただし、ある瞬間のデータが欠落することを懸念する必要はない。むしろ多くのデータを収集することで欠落データを補完することを考えたほうがよい

IoTデバイスの注意点

- ビーコンはなりすましが可能
  - ビーコンの識別は UUID / Major / Minor などの情報で行われるが、これらは平文で送信されている
  - 従って既存のビーコンと同じ情報を発信するクローンは容易に作れる(たとえば、iPhoneをビーコンにするアプリは AppStore で配布されている)

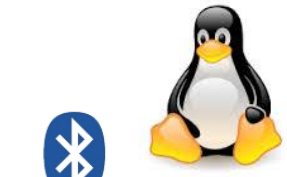


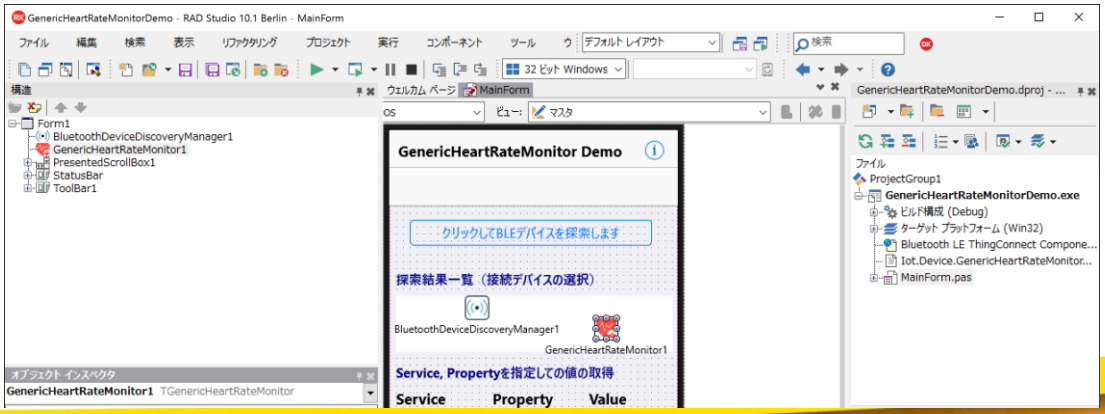


今後の展開とまとめ

■ 今後のロードマップ

- IoTデバイスサポートの強化
  - より専門的なIoTデバイスとの接続性を提供
    - 例えば「心拍計コンポーネント」「血圧計コンポーネント」など
  - IoTデバイスとバックエンドシステム間の調整を行う機能
    - エッジウェア（取得したデータのフィルタリング、バッファなど）
- 中間サーバー機能をLinuxでも実装可能に
  - マルチデバイス対応のターゲットプラットフォームにLinuxを追加
  - サーバサイドアプリの動作環境としてLinuxを選択可能に





まとめ

- IoTでは様々な情報が人手を介さずに入手可能に
  - 自動的かつリアルタイムの情報収集が可能
  - 収集した情報の分析により、現状の問題点の調査や今後の予測などに利用可能
- IoTデバイス特有の注意事項に気を付ける
  - BLEはペアリング不要かつ平文通信
  - ビーコンで近接検知や距離や位置の測定が可能
  - ビーコンのなりすましに注意（偽ビーコンはカンタンに作れる）
- IoTデバイスからのデータ収集アプリは Delphi で作れる
  - RAD Studio の IoT コンポーネントやBLEコンポーネントを利用すれば、Delphi のアプリケーションに IoT デバイスとの通信機能を追加できる
  - EMS / RAD Serverとの連携によりデータベースへのデータ収集も行える