

2019/11/1 JEITA組込み系ソフトウェア・ワークショップ2019
「DX時代を生き抜くIoTソフトウェア開発を知る」

空調機の保守サービスを支えるIoTシステム開発

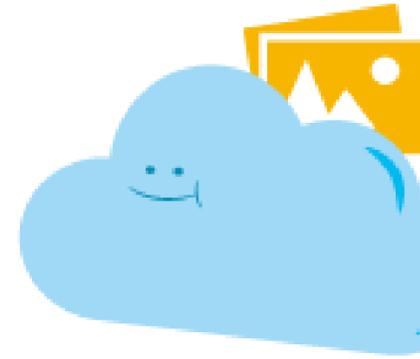
～機械メーカーがどうやってIoTを利用するのか、可能性と現実～

ダイキン工業株式会社
空調生産本部
橋本 雅文

機器組込み (ハードウェア・ソフトウェア)



サーバー
WEBアプリケーション
データベース

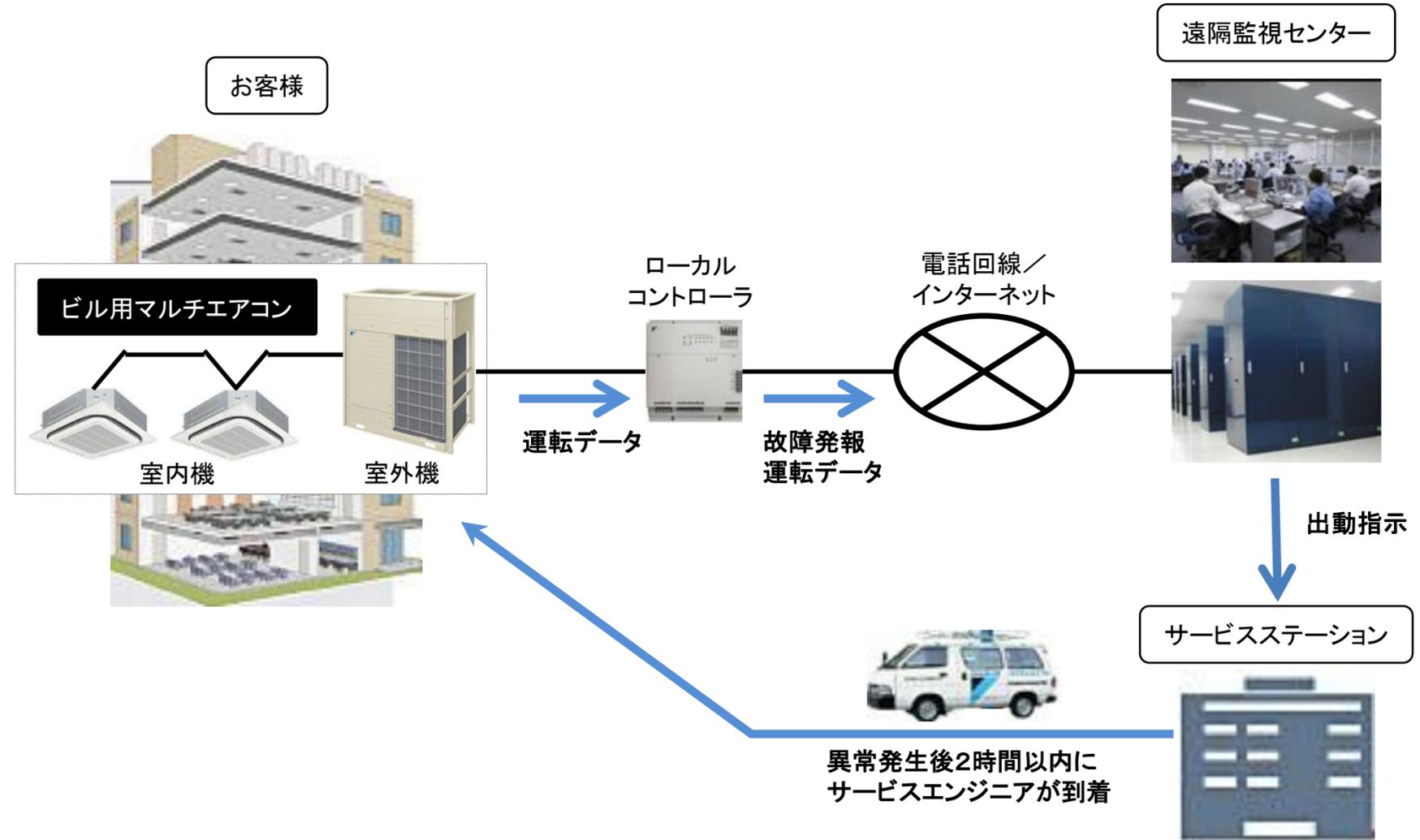


端末アプリケーション



ダイキンが開発したオンライン診断システムで空調機を遠隔診断して、異常停止や突発修理などのトラブルを未然に防止。

- 室内機では熱交換器やエアフィルターの汚れ、室外機では送風機や圧縮機の異常、冷媒漏れなどをチェック。
- 万一の故障の際は2時間以内にエンジニアが緊急出動。
- 故障予知によって機器の汚れなどによる過大負荷を抑制し、ムダな電力の消費を抑える。



1993年にサービス開始。
アナログ電話回線 + TCP / IP
1日1回データ送信

ローカルコントローラで空調機の日データを収集し、故障の兆候を常に監視。故障の兆候があれば直ちに監視センターへ通報し、サービスエンジニアは直ちに現場へ出動。

遠隔監視型の空調メンテナンスから遠隔型の省エネマネジメントサービスへ

- 1993年 空調メンテナンス『エアネットサービス』発売
- 1996年 ビルメンテナンスサービス『ビルエアネット』発売
- 1998年 ビル統合監視盤『D-BIPS』発売
- 2006年～ 省エネマネジメントサービスを順次発売

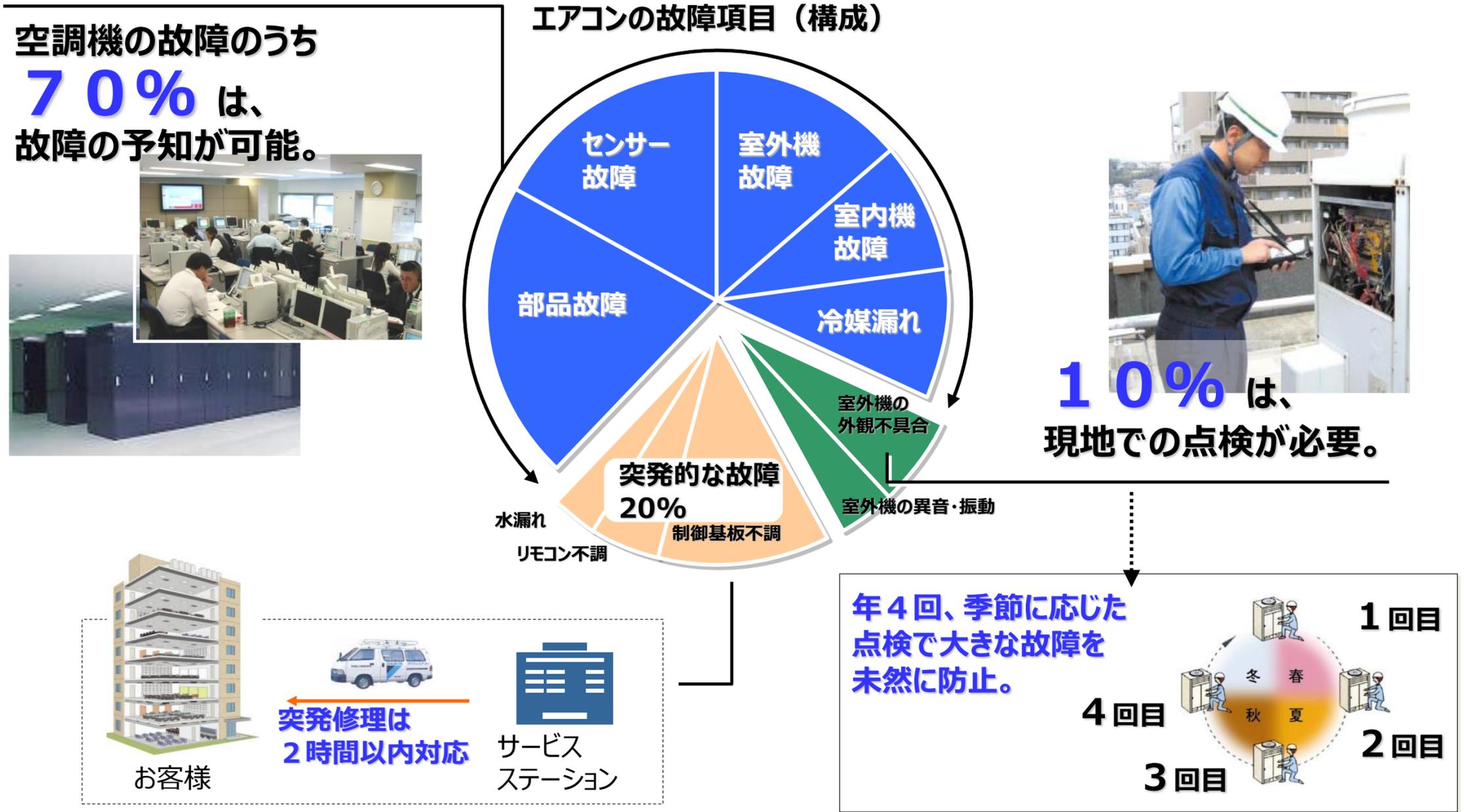
'93 (H5)	'94 (H6)	'96 (H8)	'98 (H10)	06 (H18)	08 (H20)	10 (H22)
<p>秋 エアネット発売 異常監視・稼動 空調全機種</p>	<p>春 メンテナンス会社設立 秋 オンライン診断 ビル用マルチ</p>	<p>春 機種拡充 〔吸収式 エナジオ ターボ冷凍機 他〕 春 ビルエアネット 発売</p>	<p>春 D-BIPS 発売 夏 修理無償 サービス開始</p>	<p>夏 洗浄式 高性能フィルタ 発売 秋 省エネ当番 発売</p>	<p>6月 D-BIPS IB 発売 シリーズ展開 11月 エアネット II 発売</p>	<p>4月 ENE・ FOCUS 発売 4月 性能当番 発売</p>

エアネットコントロールセンター

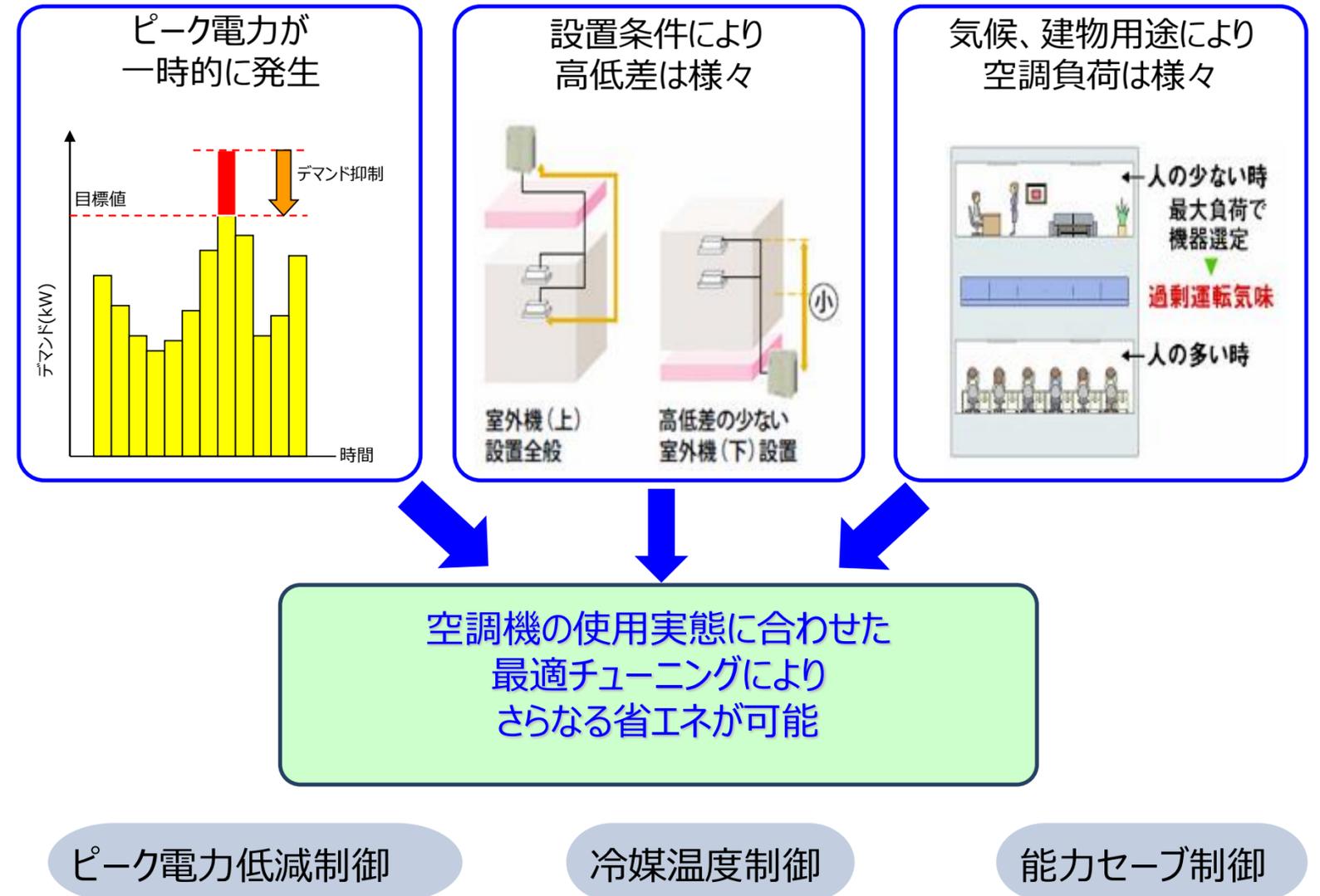
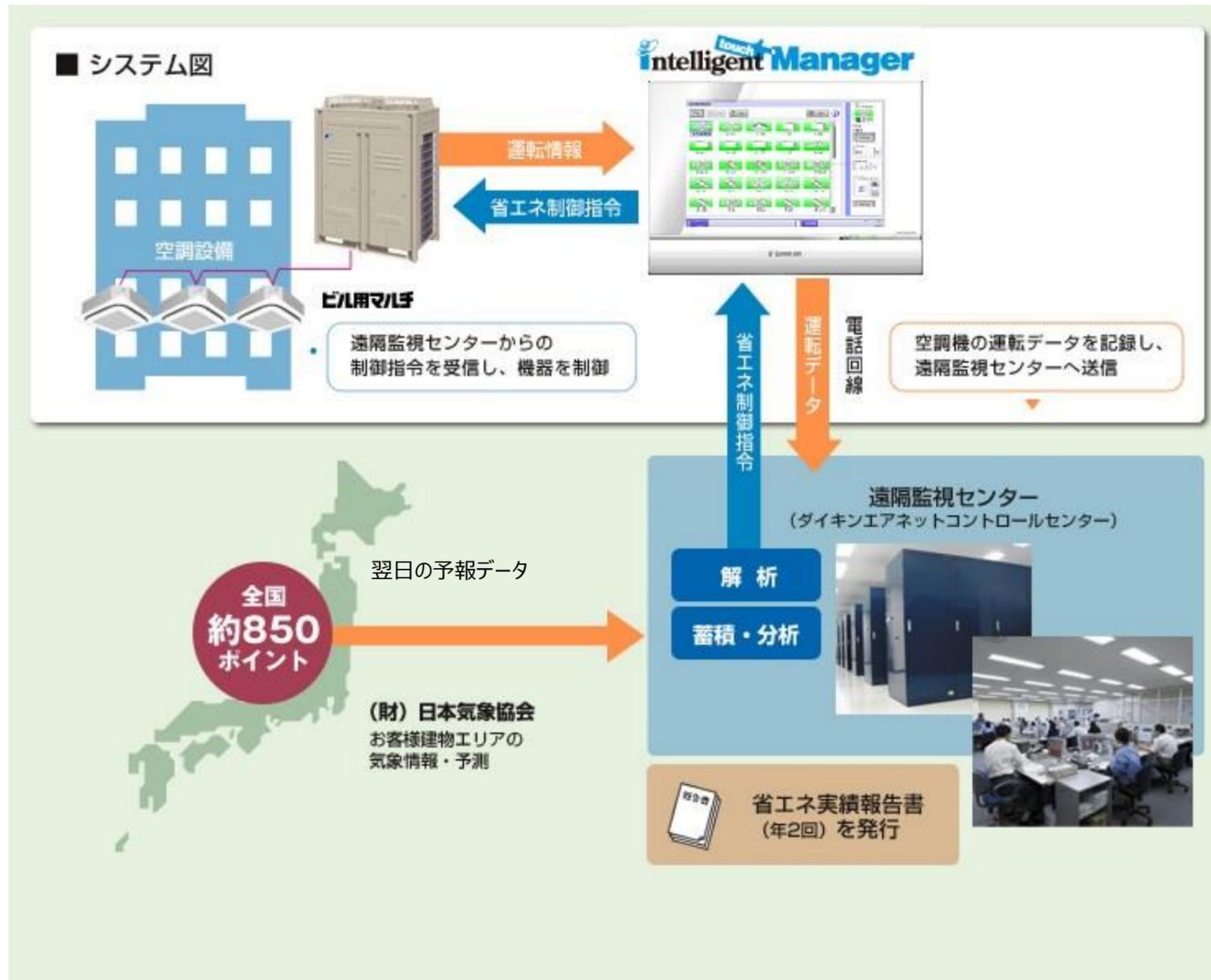
2012(H24)～
エアネット i 発売

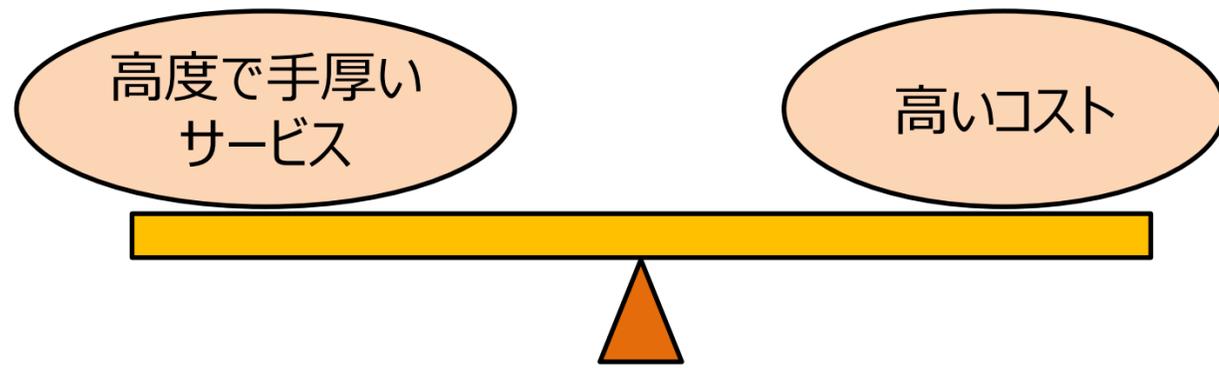
2014(H26)～
エアネットセレクト発売

遠隔監視による故障予知に加えて、年4回の現地点検で空調機故障の約80%未然に予防



空調機の運用状況とお客さまビル周辺の気象予測から、空調機を「省エネチューニング」、
空調機の電気代を最大：約20%削減 ※当社試算 2000m²クラス、60HZ地区の事務所ビル



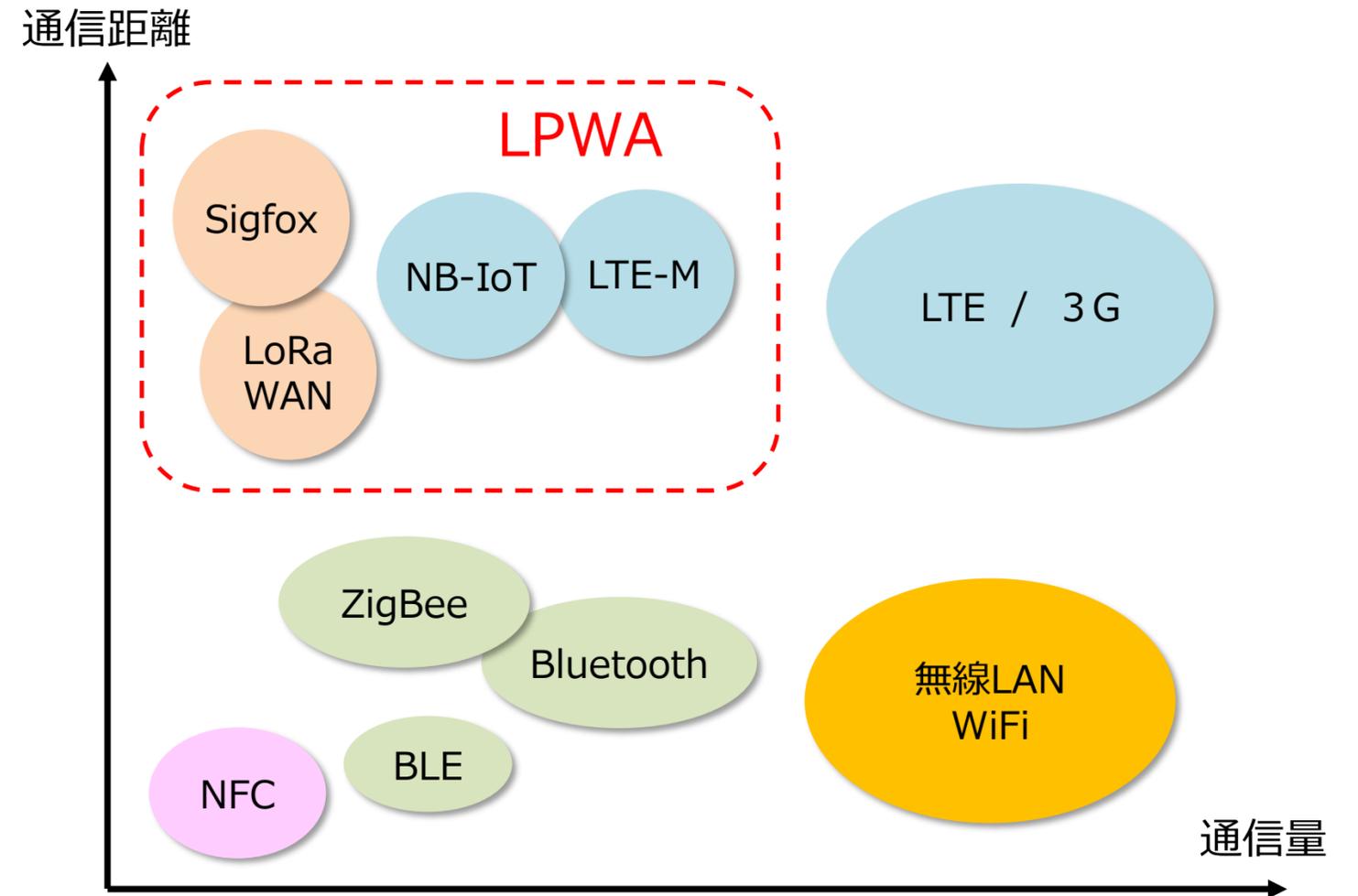
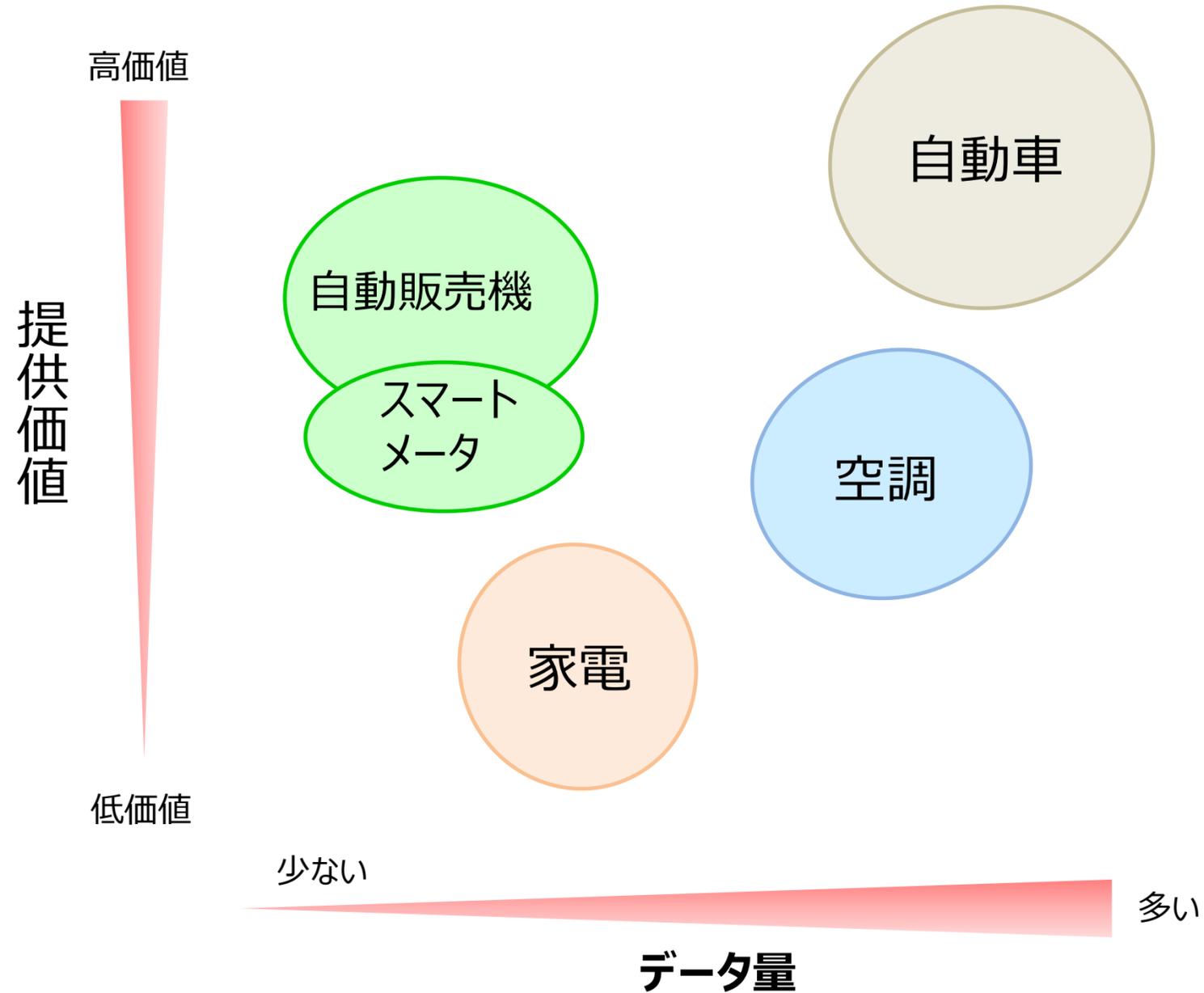


1. 施工コスト
ネットワーク工事、回線工事
空調機のアドレス設定
2. ランニングコスト
通信費、サーバー費、システム運用費
3. 通信装置コスト
データ処理装置、通信モデム



サービスを広げていくために、コストは重要。

業務用空調機は、データ量と提供価値が中間的。
 価値に見合うコストのバランスが難しい。L P W A は相性がよい。



項目	LTE/3G	WiFi	LPWA
消費電力	×	×	◎
データ量	◎	◎	△
通信コスト	×	◎	○
通信ICコスト	×	◎	○
ダイレクト接続	○	×	○

シンプルなサービスをリーズナブルに提供

2018年11月サービス開始

アンシン!
異常お知らせメール

カンタン!
フロン点検・管理

バンゼン!
空調機保全計画サポート

ダイキンは最新のIoT技術で 空調機管理をアシスト



ASSISNET SERVICE

【IoT空調管理】アシスネットサービス

オトクな! 月々 **600**円/台

※現地調査費・通信機・設置価格込み（消費税別）



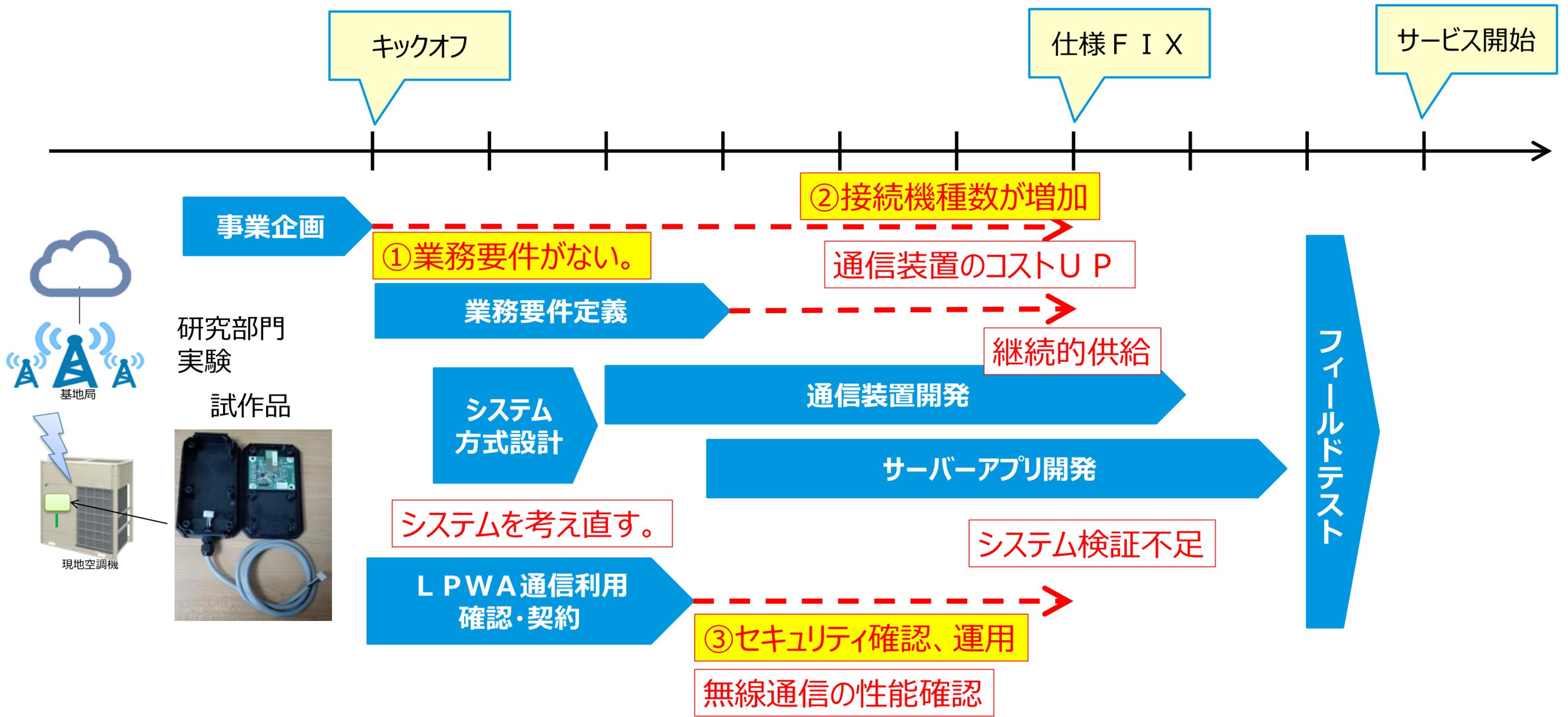
取り付けカンタン!



室外機



LPWA



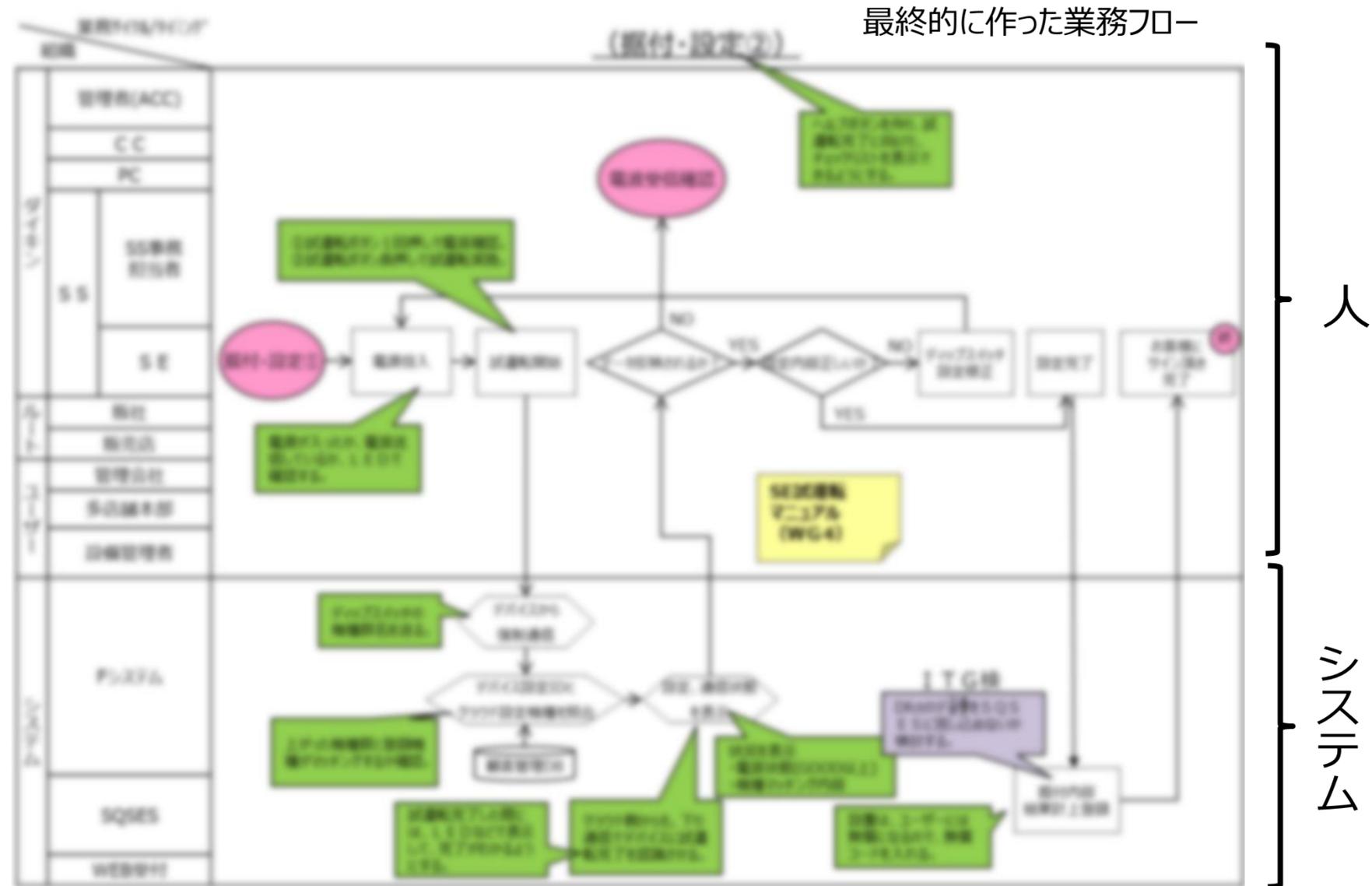
- ・事業企画（案）で、原価目標、売上目標を置いた。
- ・LPWA通信が空調機に接続して動くことを技術的に確認した。



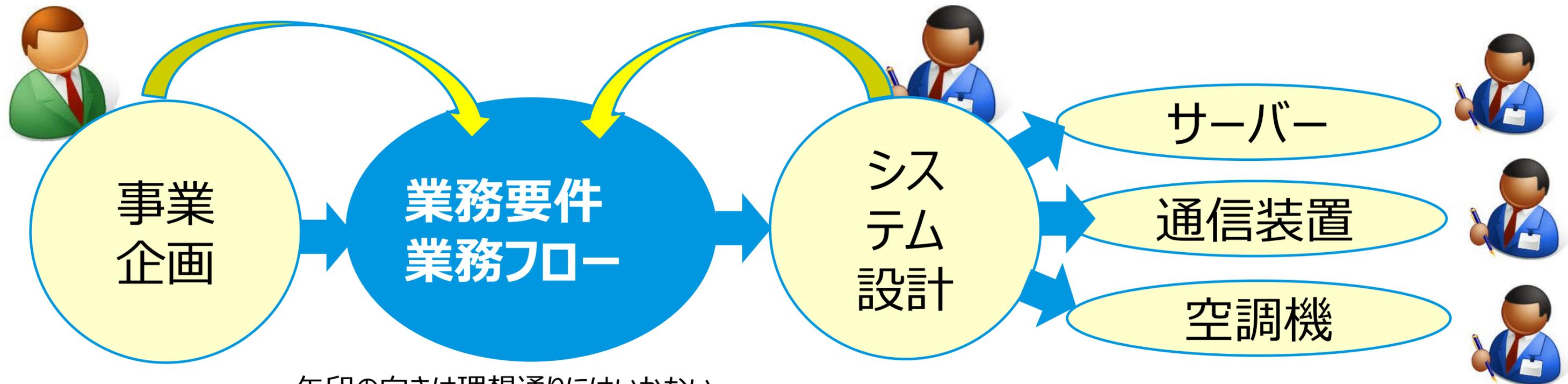
コスト試算して収益性を評価した上で、
計画を立ててすすめよ。GO！

そもそも、このサービスでは、
IoTを使って誰が何をやるの？
がわからないと・・・
コストも開発計画も立てれない。

業務フローをつくろう！どうやるの？
事業企画者、機器組込みソフト設計者は知らない。
勉強しながら一緒に業務フローを作った。



空調機をわかっていて、通信をわかっていて、セキュリティをわかっていて、サーバーをわかっている人が、事業企画者と一緒に作業しないと、IoTを利用したビジネスの業務フローをつくることができない。



矢印の向きは理想通りにはいかない。

サーバー、通信装置、空調機、そして現場の作業を全てをわかっている人はいない。
 サービスマン、空調機設計者、通信設計者が協議することで、業務要件が足されていく。
 開発の終盤に、業務要件が増えることも・・・

企画時（企画時）

- ・フロン点検法（法律）の対象機器を、I o Tの接続対象にした。
- ・ビル用エアコンを対象にした。
- ・既に販売した機器も対象にした。（15年前に販売した機器まで対象）



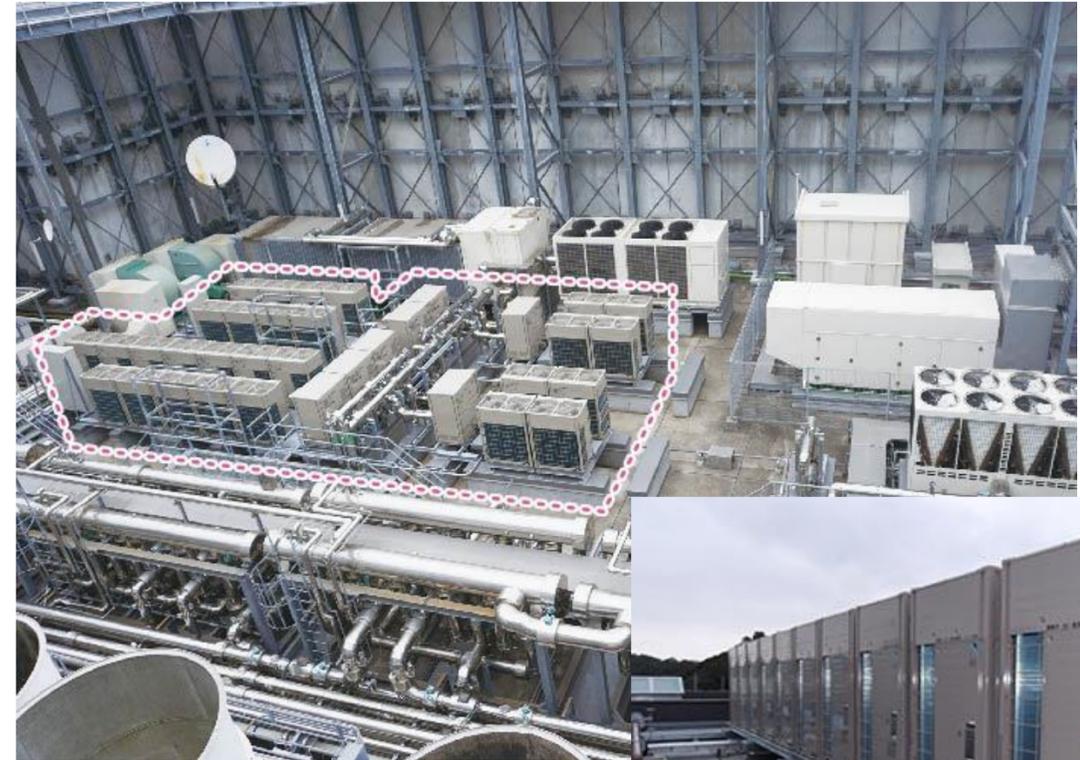
- ・販売店
 - ・サービスステーション
 - ・ベテラン社員
- に聞いていくと・・・



お客様の建物に
設置している
当社の空調機は全て
つながらないと
使ってもらえない。



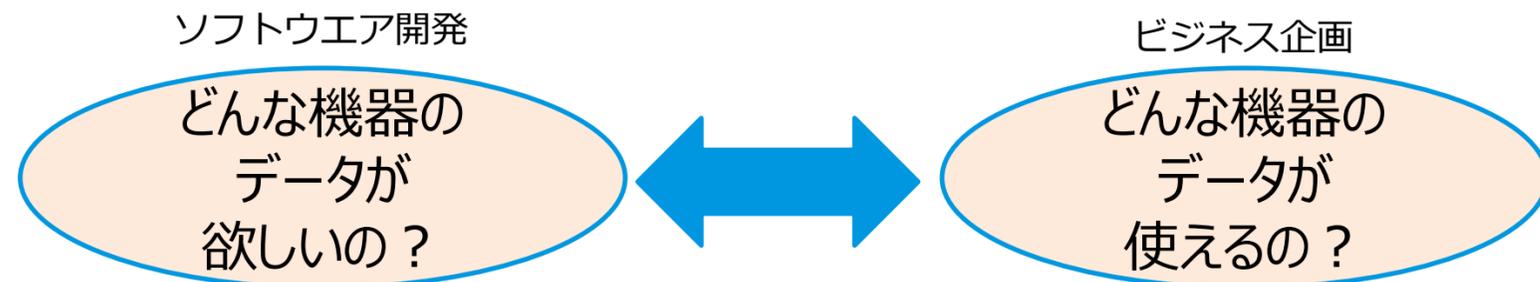
接続する対象機種数：約700機種
結果、ビル用だけでなく店舗用も対象に。



モノをネットワークにつなげる I o T では、接続する機器の仕様限定はとても大切。
 しかし、究極の要求は、“**なんでもつないでほしい**” ということ。

通信装置を介して、機器をネットワークにつなぐだけならまだ簡単。
 サーバーアプリケーションに必要な機器のデータをそろえて、使えるようにするのが難しい。

そもそも、機器組込みソフトウェアの設計者は
 を機器をネットワークにつなげて
 データを使えるように設計していない。



データ名	ビル用エアコン 1世代前(v5)				ビル用エアコン 最新シリーズ(v6)				ビル用エアコン 冷暖フリーシリーズ (v4R)				店舗用エアコン R410A (スカイエアR410A)				店舗用エアコン R32 (スカイエアR32)			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
5	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
6	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
7	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
9	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
10	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
11	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
12	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
13	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
14	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
15	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
16	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
17	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
18	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
19	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
20	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
21	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
22	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
23	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
24	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
25	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
26	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
27	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
28	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
29	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
30	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
31	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
32	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
33	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
34	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
35	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
36	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
37	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
38	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
39	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
40	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
41	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
42	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
43	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
44	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
45	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
46	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
47	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
48	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
49	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
50	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

機器に設置した L P W A 通信装置とサーバー間は、通信サービスを利用した。
セキュリティは、全部通信サービスにお任せした。・・・と考えていた。

しかし、実際に運用を考えていくと
疑問がいっぱい。

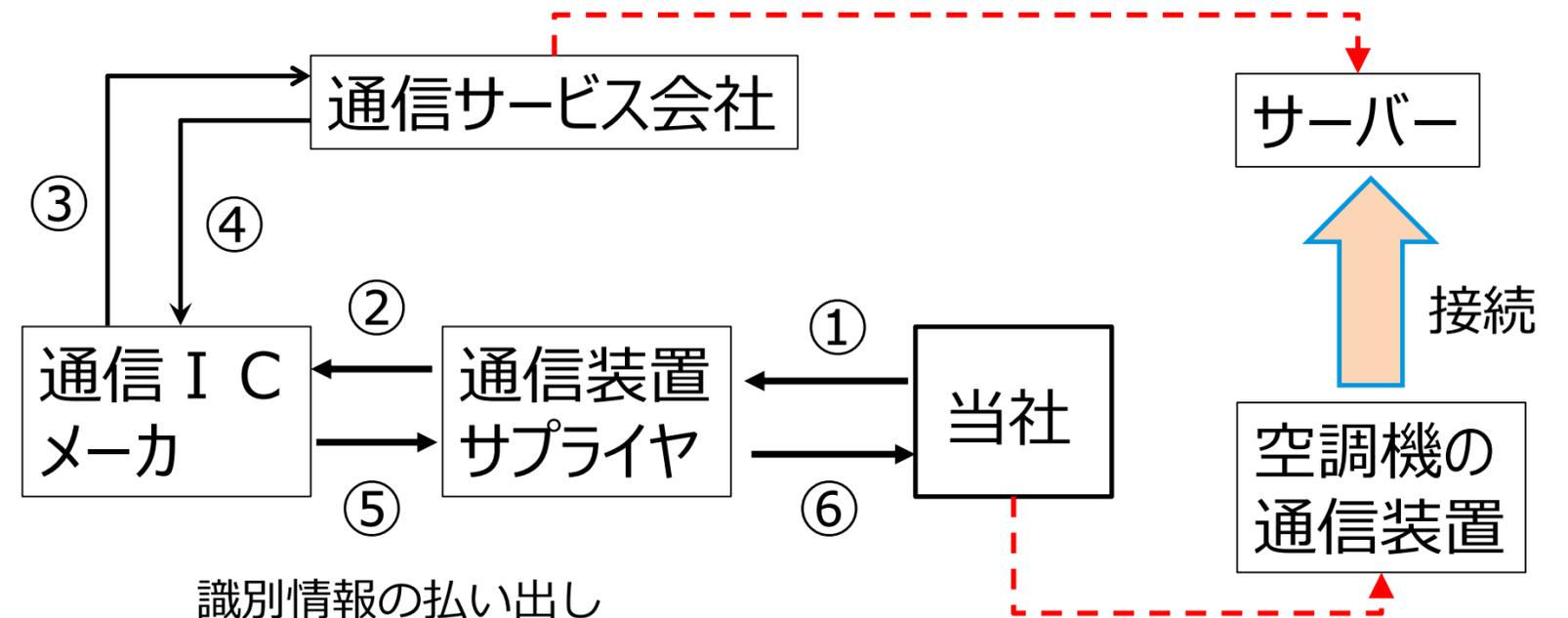
- 通信サービスの転送データの暗号の強度は大丈夫？
- 通信 I C に書き込む個体識別の I D は、誰が払い出すの？ 誰が管理するの？ どうやってやりとりするの？
- 通信 I C の個体識別 I D はサーバーにいつ登録されるの？

聞いてみた。



なかなか、よくわからなかった！
実験的にはやっていたが・・・
量産となると・・・

- サービス利用時に、個体識別 I D をどうやって接続作業者に教えるの？



- セキュリティの要件を、実システムの各アイテムに展開することの難しさ。
- 通信装置などモノの調達、セキュリティの重要情報管理など、継続的に運用することまで考える。
- 運用はセキュリティに限らない。ソフトウェア保守（OSS、クラウドサービスやOS更新）も・・・

セキュリティ要件

- ネットワークにつなぐ機器（通信装置）は個体識別すること
- 転送データは暗号化すること
- 認証情報、鍵の管理を行うこと。（第三者が確認できること）
- 通信ログを残すこと。など・・・

実システムの各アイテム
(組織、モノ) に要件を展開

- 通信装置
(基板設計・ソフト設計)
- サーバー
(ソフト設計・インフラ設定)
- サプライヤ
- 調達部門
- 事業部門
- 現地工事者

設計
開発

モノ
生産

運用

ソフト
保守

メーカー
得意

誰が展開するのか？

メーカーの習性

1. モノをつかって動いたら、ヨシ売ろう！となる
2. モノは売ったら、手元から離れる。品質をしっかりと作りこむ。運用って？

モノの開発

モノが提供する品質（機能、性能）
を目標においたモノのデザインである

- ・モノの目標値は数値化しやすい
- ・目標値から要求品質展開
- ・部品、部材、組立て = 原価



I o T 開発

モノがつながる仕組みを利用した
業務デザインである。業務 = 人の行い

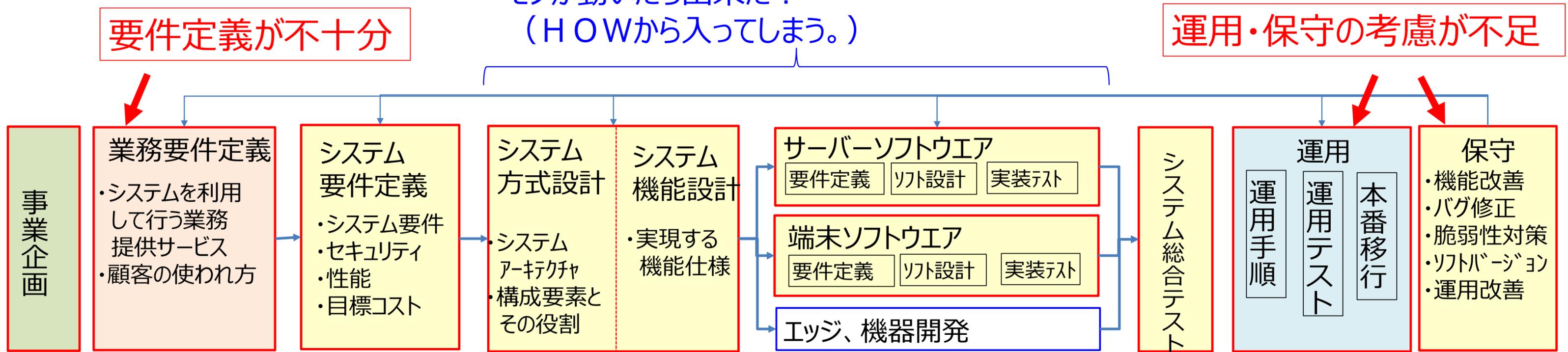
- ・モノ、人、情報で やりたい事が目標？
- ・初めに、全ての課題展開は困難。
- ・モノ、ソフト開発、人の工数 = 原価？

世界観が違う



- ・モノづくりは、テクニカルな課題から始めてしまう。
- ・人、モノ、情報 の要件定義が大切。業務フローをつくらないと実際の開発は始まらない。
- ・全体をわかって要件定義をできる人はいない。
待っていても要件は定義されない。ソフトウェア設計者の役割は重要。
- ・システム運用、ソフトウェア保守の誰がやるのか・最初に決めない。

テクニカルなところから始めてしまう。
モノが動いたら出来た！
(HOWから入ってしまう。)



参考：【ソフトウェアライフサイクルプロセス IEC・ISO12207】共通フレームワーク2013

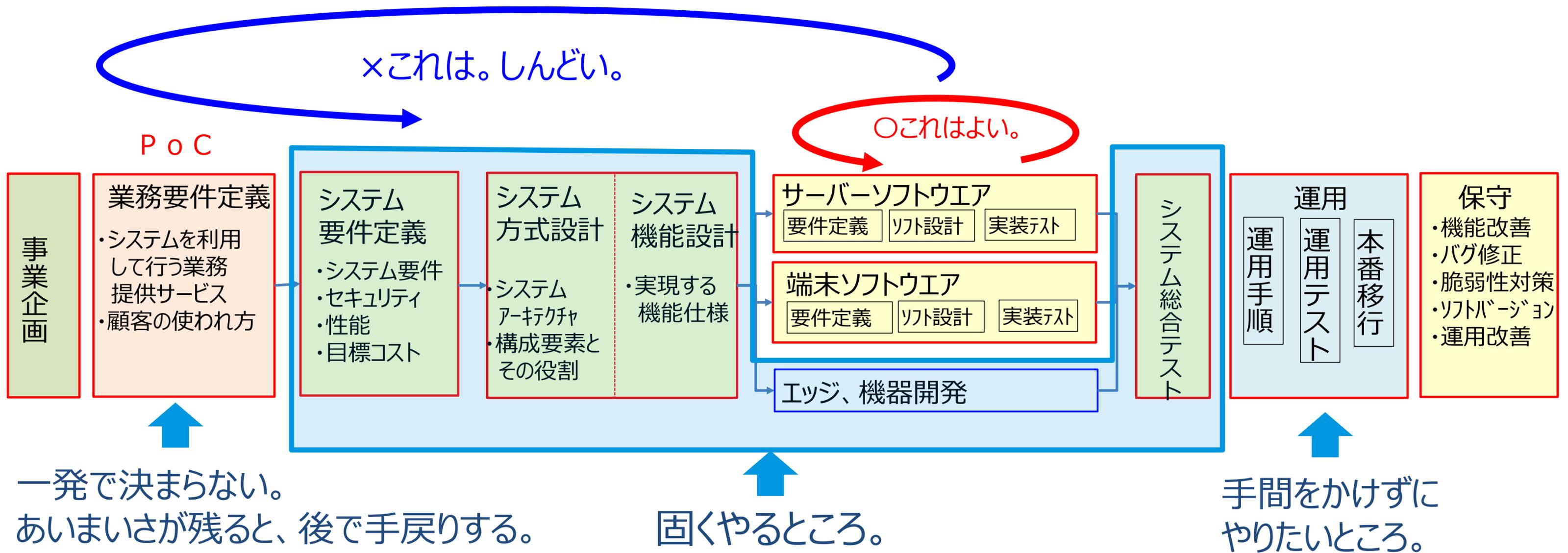
•IoTは、試して繰り返して仕様を決めたいが...

◆システム要件・設計を繰り返すとしんどい。

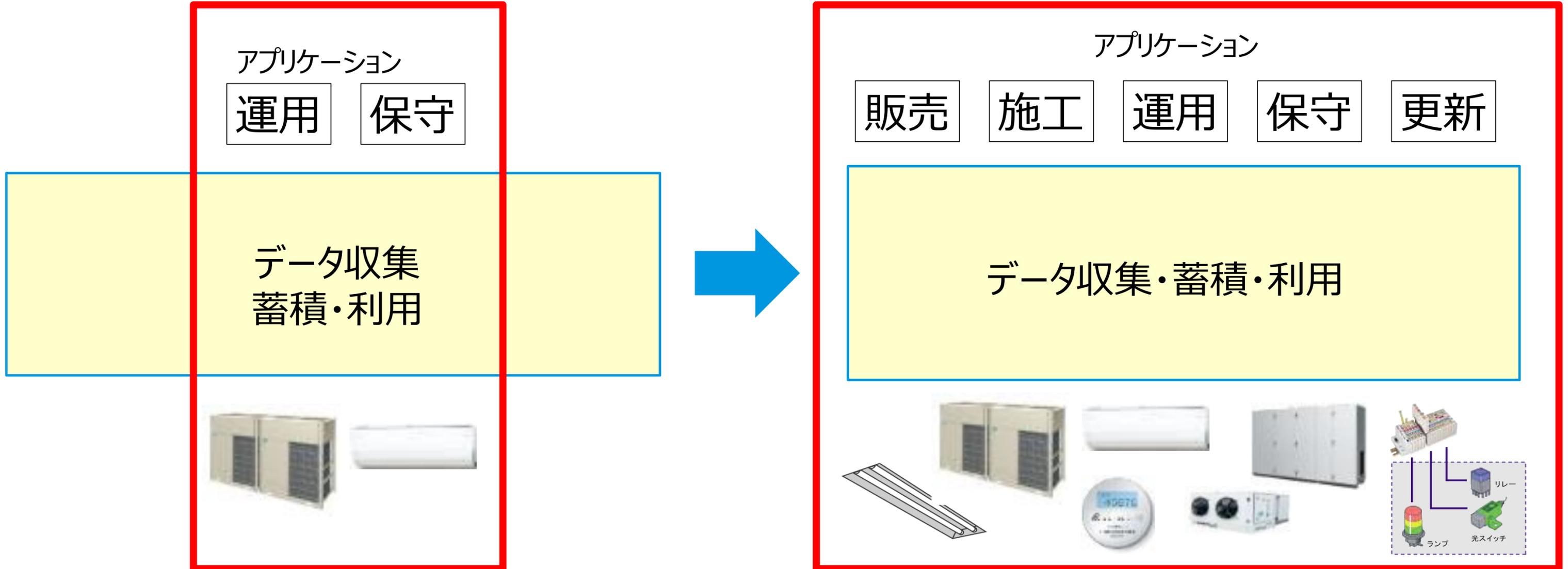
◆人が使うアプリケーションは、もっと使いやすく。アジャイル？

◆最初の業務要件定義が決まらない。PoC？

◆開発したソフトウェアをスムーズに運用へ。



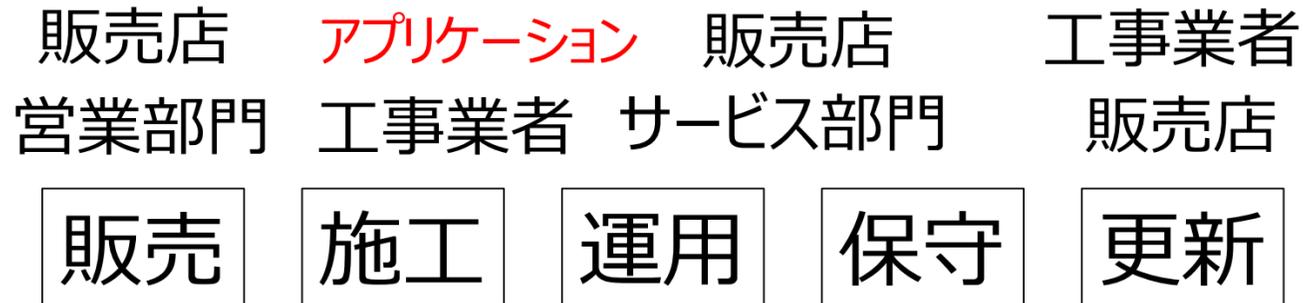
- ・現状、まだまだシンプル。空調機中心で、運用・保守のアプリケーション。それでも大変。
- ・ビジネスのライフサイクルで、様々なモノが繋がっていく。（つなげたい・・・）



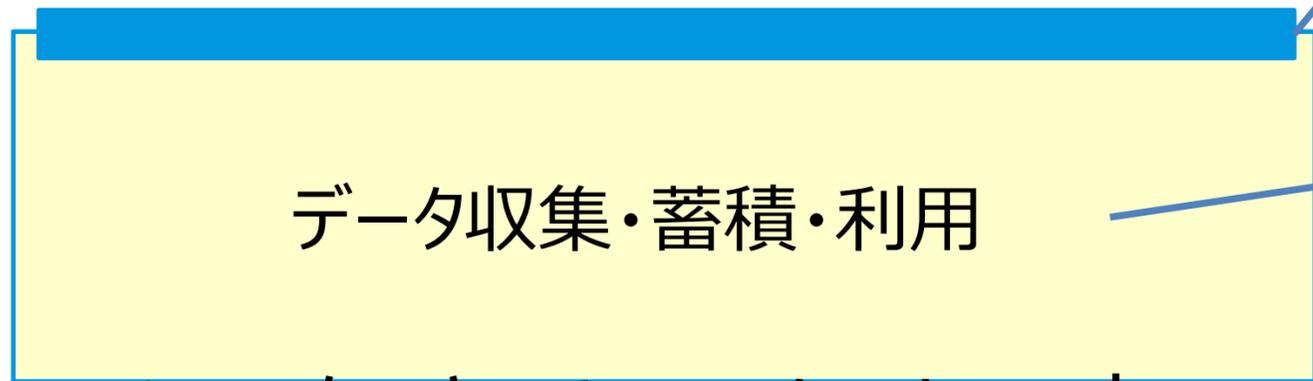
現状は部分的、まだまだ シンプル！

ライフサイクル全体で最適なサービス
様々なモノがつながる。

- ・アプリケーションと機器は、同時並行・多発的に設計する。
- ・ソフトウェア開発を破綻させないために、「つなぎ方」「データ形式」「A P I 設計」は重要。



API設計
アプリケーションができる事を制約し、機器の詳細仕様を知らなくても設計できる。
機器の仕様差が不具合になる。



データ形式
アプリケーションが扱えるようにデータの形と定義をそろえる。
アプリケーションは後から増える。



つなぎ方
できるだけ、接続方法の種類は減らす。
簡単につなぐ = 専用化
色々な機器をつなぐ = 汎用化 } 矛盾

なんでもつなげて、なんでもできるプラットフォームは存在しない。

1. 「IoTは、業務デザインである」ことへの理解。
技術を開発して、モノに詰め込んで生産して売ると違う。
業務の困り事解決や生産性向上を提供する。実際の開発は行ったり来たりする。
2. IoTの開発は、様々な要素が関連するシステムとアプリケーション開発。
“従来型の重いプロセス” と “軽く回すプロセス” の両方が組み合わさる。
業務要件は、開発過程において最後まで修正が加わる。
業務要件に応える固いシステム + ライトな多数のアプリケーション構成。
3. IoT開発は多領域にわたる。技術者のフルスタックエンジニア化。
 - ・企画、営業の人も、IoT技術、ソフト開発プロセスを知ってもらう。
 - ・組込み系エンジニアも、現場やビジネスを知ってもらう。
 - ・自分の専門知識 “プラス1” の知識を学ぶ。

空気で答えを出す会社

