

派生開発カンファレンス 2025 ワークショップ

サイバーフィジカル統合サービスのための
XDDPプロセス構築法
～スマートロック QR機能開発事例から学ぶ～

2025/05/23(金) 10:00-11:45

派生開発推進協議会 T6/T4研究会

目次

1. はじめに
2. XDDP概要(派生開発におけるメリット紹介)
3. 方法論解説
4. 課題説明
5. グループワークと発表
6. 解説・まとめとQ&A

1. はじめに

- スマートロックシステムへのQRコード読取機能追加事例を通じて、XDDPを企業間/ Webサービスレベルに拡張したエンタープライズXDDP(以下**eXDDP**)の実践手法とUSDM/TMを活用した派生開発の効率化を学ぶ
- 実際に課題に取り組みながら、サイバーフィジカル統合サービスにおける開発期間短縮方法を体験
- **ワークショップの課題**
 - **前半: スマートロック QRコード認証の追加プロセスを説明**
 - **後半: 上記に顔認証機能を追加する eXDDPを体験**
- 対象
 - 対象者: システム開発担当者、プロジェクトマネージャー、製品管理者
 - 所要時間: 105分
 - 参加人数: 5-6人グループに分けて実施

2. XDDP概要(派生開発におけるメリット紹介)

- 清水吉男氏の提唱したXDDP (eXtreme Derivative Development Process) は派生開発の QCD (Quality, Cost and Delivery) を向上させる有効なプロセス
- AFFORDDのXDDPの特徴を元にXDDPとeXDDPとの違いについて説明
- XDDPの特徴
 1. 変更と追加プロセスを別々のプロセスで対応する
 2. 「変更要求」と「変更仕様」を階層構造の中で扱う
 3. 変更仕様をソースコードのレベルで捉える
 4. 基本的に「差分」で作業を進める
 5. 「変更3点セット」の成果物で協同作業を円滑にする



2.1 変更と追加プロセスを別々のプロセスで対応する

- 清水吉男氏の説明は以下の通り。次シートも同じ構成で違いを説明
- 派生開発には「変更の要求」と「機能追加の要求」がある。変更の要求は、現状すなわちbeforeの仕様をafterの仕様に変更する方法で実現するが、機能追加の要求は新しい機能の要求仕様を作成し、設計し実装するというステップで実現する。つまり両者の要求の性質は全く異なる
- このため、XDDPでは「変更要求」を実現するプロセスと「機能を追加する要求」を実現するプロセスを別々のプロセスで対応する。これによって機能追加の要求を実現するために、**ベースのソースコードをどのように変更して実現するかを、「変更要求仕様」を対応させる方法で扱うことができる**
- **★eXDDPにてにおいても before/afterの変化点は、サブシステムレベルでの IFの変更**に現れてくるという考え方から「変更要求仕様」に記載すると考える

2.2「変更要求」と「変更仕様」を階層構造の中で扱う

- 「USDM(Universal Specification Describing Manner)」では「仕様は要求の中の動詞にある」というのが基本的な考えであり、この考えの下で「要求」と「仕様」を階層構造で表現して仕様を漏れなく抽出する
- XDDPでは、この考え方を「変更」の世界に適用します。つまり、「変更要求の中で表現された”変更する”という行為に変更仕様がある」と考えることで、具体的な変更箇所を抽出しやすくする
- **★eXDDPでも同様に、変更箇所を抽出するが、振る舞いとサブシステムでの IFの変更に着目する**

2.3 変更仕様をソースコードのレベルで捉える

- 新規開発や機能追加では、そこで求められている「仕様」を実現する方法はこの後の「設計プロセス」の中で考えられ、レビューのようにその方法が適切かどうかを評価する機会があるが、変更には通常は「設計プロセス」がなく、変更の実現方法を評価するタイミングがない
- そのため、一般には最後にソースコードレビューを行っているが、その効果は限定的なものに成らざるを得ない
- **XDDPでは、変更仕様としてソースコードのレベルで記述するが、そのことによって早い段階で変更の実現方法を評価する機会を確保し、より適切な変更方法の選択を可能にする**
- **★eXDDPでは、変更仕様として IF の変更にともなうサブシステムレベルでの振る舞いとテスト箇所を定義することで、受入テストの内容を記載し合意することでレビューとする**

2.4 基本的に「差分」で作業を進める

- ベースのソースコードや、これに対応する機能仕様書や各種の設計書は「公式文書」であるにもかかわらず、一般にはこれらの公式文書を変更しながら作業を進めていて、このことが多くの混乱の元にもなっている
- **XDDPでは、変更作業は原則として「差分」で進めていく。ソースコードの変更も事前に差分情報を揃えておいて一気に作業に着手し、1回で変更作業が終了する。その結果、ソースコードの変更し直しがなく、必要最小限の工数でソースコードの変更が可能となる。なお公式文書の更新はテストで変更の正しさが確認されてから行う**
- **★eXDDPでは、変更作業は「差分」の before/afterに着目して進む**
- **変更箇所は他社のクラウドサービスでソースコードまで開示されない場合もあるため、変化点に着目した受入テスト完了後に、マニュアルなどの公式文書を更新する**

2.5「変更3点セット」の成果物で協同作業を円滑にする

- 変更する箇所の情報を「変更要求仕様書」「TM(Traceability Matrix)」「変更設計書」の「変更3点セット」の成果物に記述することで、各担当者が予定している変更内容や変更箇所、具体的な変更方法をお互いに知ることができ、早い段階で調整することができる。しかもこの時点では、ソースコードを含めて公式文書は一切変更していないので、各担当者が参照する情報に混乱は生じません。**さらに担当者が変更設計書を揃えて一気にソースコードを変更することで、結合後の混乱もほとんどありません。**
- **一般に行われているような各担当者が該当する箇所を見付け次第に変更してしまう方法では、他の担当者が自分に影響する情報を事前に知る方法がなく、最後にそれぞれの担当範囲を合わせてテストしたところで変更箇所の食い違いや考え方の違いが一気に噴き出す**
- **★eXDDPではシステム間での差分について受入テストレベルで合意してから着手するため統合時の混乱もほとんど発生しない**

2.6「変更三点セット」の読み替え内容

参考としてeXDDPの成果物と技法について定義と読み替え内容を示す
 (『[困ってませんか？ 派生開発 ～XDDP はじめの一步～](#)』から引用・変更)

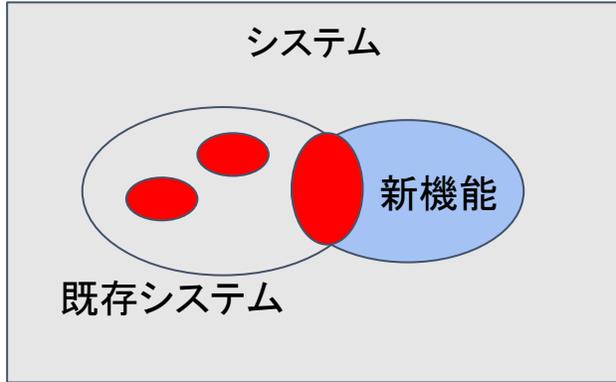
#	項目	内容
1	変更要求仕様書 (USDM)	要求仕様の表記法。要求と仕様を階層的に表現する。 また、仕様の抜けや漏れを防ぐことを目的として、要求の背景としての「理由」(why/reason)を記述する
2	トレーサビリティマトリクス(TM)	USDMと システム (変更設計書)とのトレーサビリティを確保するためのマトリクス形式文書
3	変更設計書	派生開発において システムの 変更箇所を記述した文書。原則として、TMのマトリクス交点にマークした箇所に対応して作成する。 受入テストの内容も記載する
4	プロセスフローダイアグラム(PFD)	プロセスと成果物の連鎖を記述する表記法。「プロセス」と「成果物」の連鎖を「フロー」でつないで表現する

2.7 一般的なXDDPとeXDDPの比較

観点	XDDP (ソースコード中心)	eXDDP(システム/連携重視)
主な対象	ソースコードレベルの変更	ハードウェアとソフトウェアが連携するシステム全体、特にサイバーフィジカルシステム (CPS) やIoTデバイスを含む派生開発
変更の捉え方	ソースコードの差分として捉える	システム間のインターフェース変更、機能追加、ステークホルダー間の役割分担など、より広範な変更
成果物 (変更3点セット)	変更要求仕様書、トレーサビリティマトリクス、変更設計書(主にソースコードレベル)	変更要求仕様書(USDM)、トレーサビリティマトリクス(TM)、変更設計書。特にUSDMはパートナー企業との連携や要求の明確化に活用され、抽象度を上げてシステム全体を記述
適用範囲	既存ソフトウェアの機能変更・機能追加	複数の企業やチームが関わる複雑な製品・サービス開発。特に、スマートロック事例のように、ハードウェア開発パートナー、クラウドサービスPMS/チェックインシステムなど多岐にわたる連携が求められるケース
重視する点	ソースコードレベルでの品質(Quality)、コスト(Cost)、納期(Delivery)の向上。	QCD向上に加え、開発期間の大幅な短縮、複数ステークホルダー間の円滑なコミュニケーションと合意形成、システム全体の視点での問題解決。USDM/TMを活用したトレーサビリティ確保と影響範囲の明確化
プロセスの特徴	変更プロセスと追加プロセスを分離し、差分で作業を進める。	一般的なXDDPの原則を踏襲しつつ、USDMをアンカーとしてパートナー間のコミュニケーションを促進。抽象度を上げた成果物により、ハードウェアを含むシステム全体の変更にも対応。API連携のような疎結合部分と、デバイス共同開発のような密結合部分でプロセスをテーラリングする視点も含む

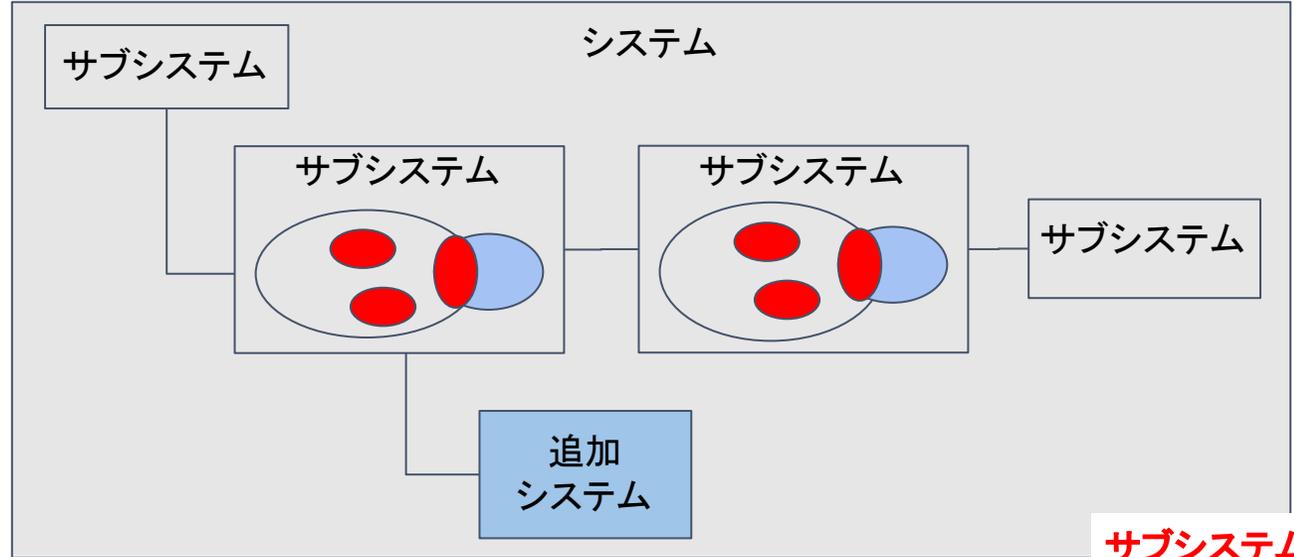
2.8 これまでのXDDPとeXDDPの比較

これまでのXDDP



赤: 変更
青: 追加

eXDDP



■ USDMで変更要求仕様書を作成する

変更要求	REQ01	目覚しの設定できる時刻を1つから5つに増やす
理由		ユーザーへのアンケートにより要望が多かったから
<時刻の設定画面の変更>		
<input type="checkbox"/>	<input type="checkbox"/>	時刻の設定の項目を1つから5つにする。
<設定時刻の保存の変更>		
<input type="checkbox"/>	<input type="checkbox"/>	設定した時刻を保存するメモリを1つから5つの配列にする。
<目覚し時刻になったときの変更>		
<input type="checkbox"/>	<input type="checkbox"/>	目覚し時刻の判断の処理で、保存したメモリ1つから5つの配列を見るよう変更する。

変更内容	変更理由	学習計画
1. 時刻設定画面の項目を1つから5つにする。	ユーザーからの要望が多いため、時刻設定の項目を増やす。	1
2. 設定した時刻を保存するメモリを1つから5つの配列にする。	時刻設定画面の項目を増やすに伴い、時刻を保存するメモリも5つ必要となる。	2
3. 目覚し時刻の判断の処理で、保存したメモリ1つから5つの配列を見るよう変更する。	時刻設定画面の項目を増やすに伴い、時刻を保存するメモリも5つ必要となる。	3

- ・変更したいことを、関係者が特定(Specify)するための文書
- ・USDMで動詞より仕様を導出し記述することで、「漏れ」や「重複」を防ぐことができる
- ・「before/after」で表現することで修正箇所や影響範囲を特定しやすくする
- ・「適切な要求の細分化」や「仕様からの適切な要求の導出」も可能

※XDDPの基本(T3研究会)から引用

振る舞いとサブシステムでのIFの変更に着目

	クラウド	メイン基板	デンソーリーダ(感振器)	センサー基板	モーター基板	金型・部品	ALNEXT	組立業者	その他
1 要求	あり	あり	あり	なし	なし	なし	なし	なし	なし
2 理由	あり	あり	あり	なし	なし	なし	なし	なし	なし
3.1 仕様	なし	なし	なし	なし	なし	なし	なし	なし	なし
3.2 仕様	なし	なし	なし	なし	なし	なし	なし	なし	なし

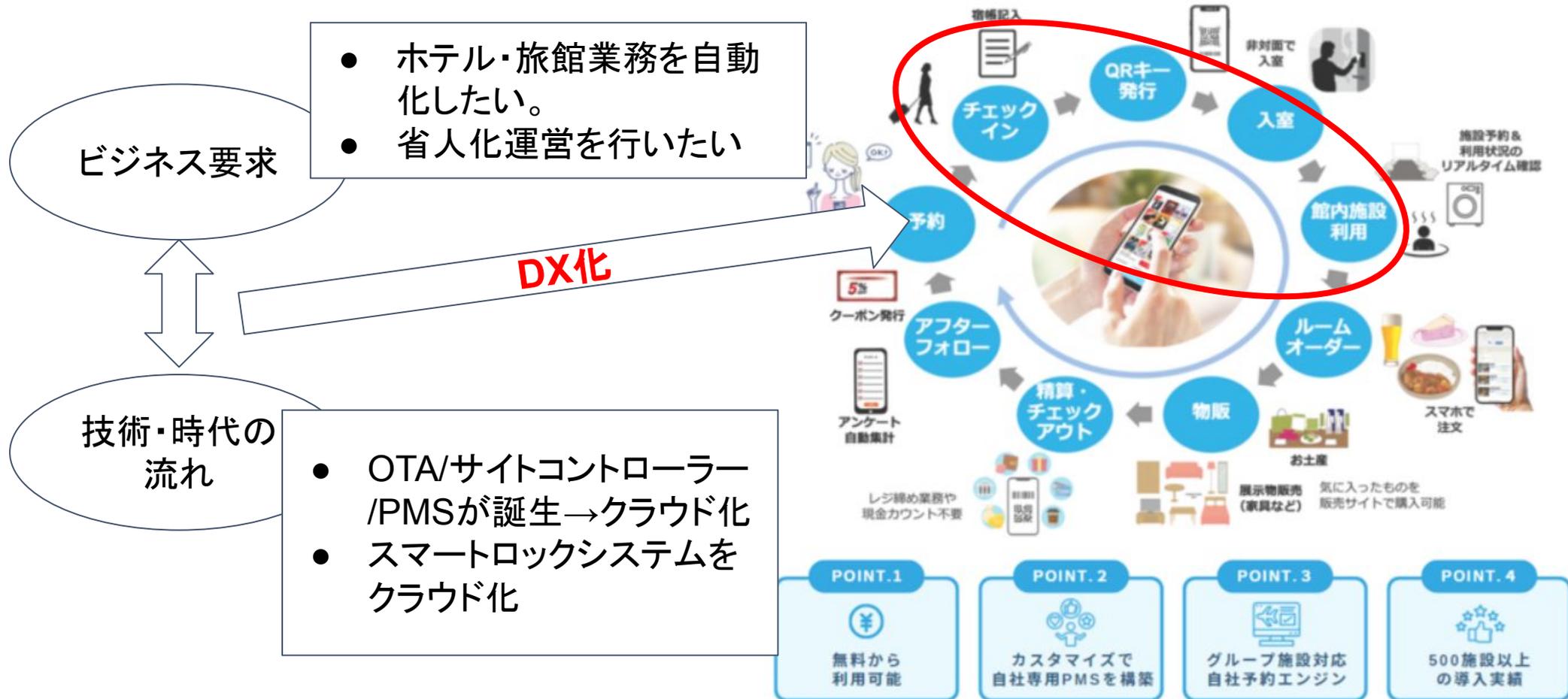
サブシステムレベルでの振る舞いとテスト箇所を定義

サブシステムレベル

3. 方法論解説

3.1 ビジネス環境変化と価値の提供について

40年かけてホテルシステム(PMS)が発展していく中でクラウドサービス化が進行、2010年代後半にスマートロックマネジメントシステムとの連携開始。チェックインの効率化・鍵の受け渡しや預かりをなくしフロントの無人化が進んでいった。(以下はRLと連携している陣屋コネクト資料から引用)



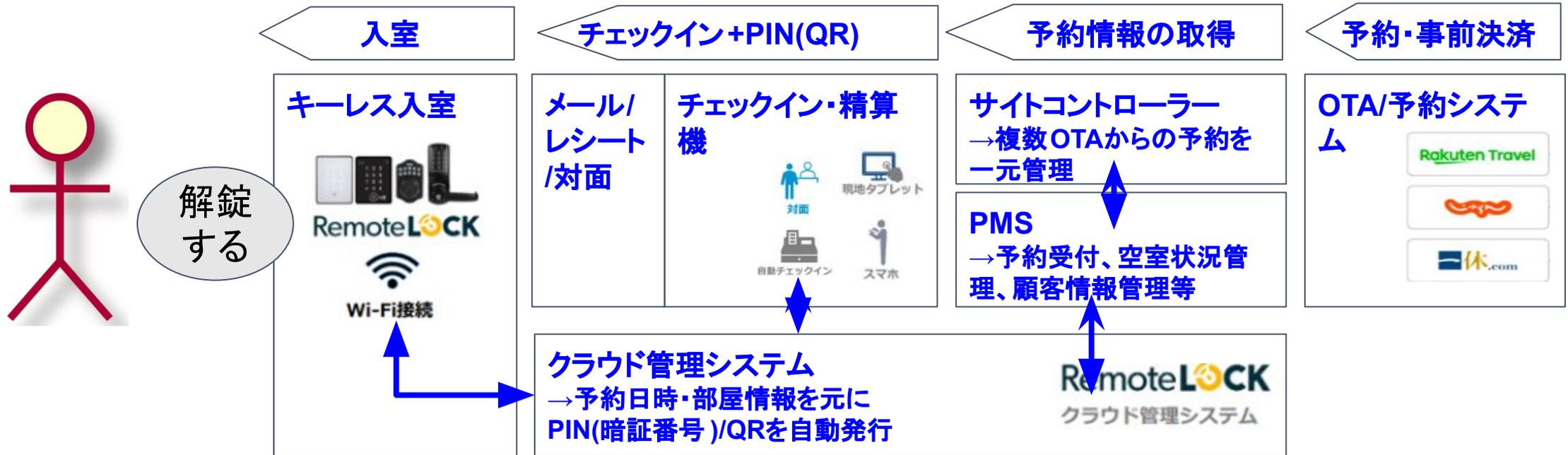
3.2 ホテルシステム連携 (これまで)

- 予約情報の流れから見ると、OTA/予約システム→サイトコントローラー→PMS→チェックイン・精算機となる
- チェックイン時にカード/物理カギを受取、客室のドアを解錠して入室する



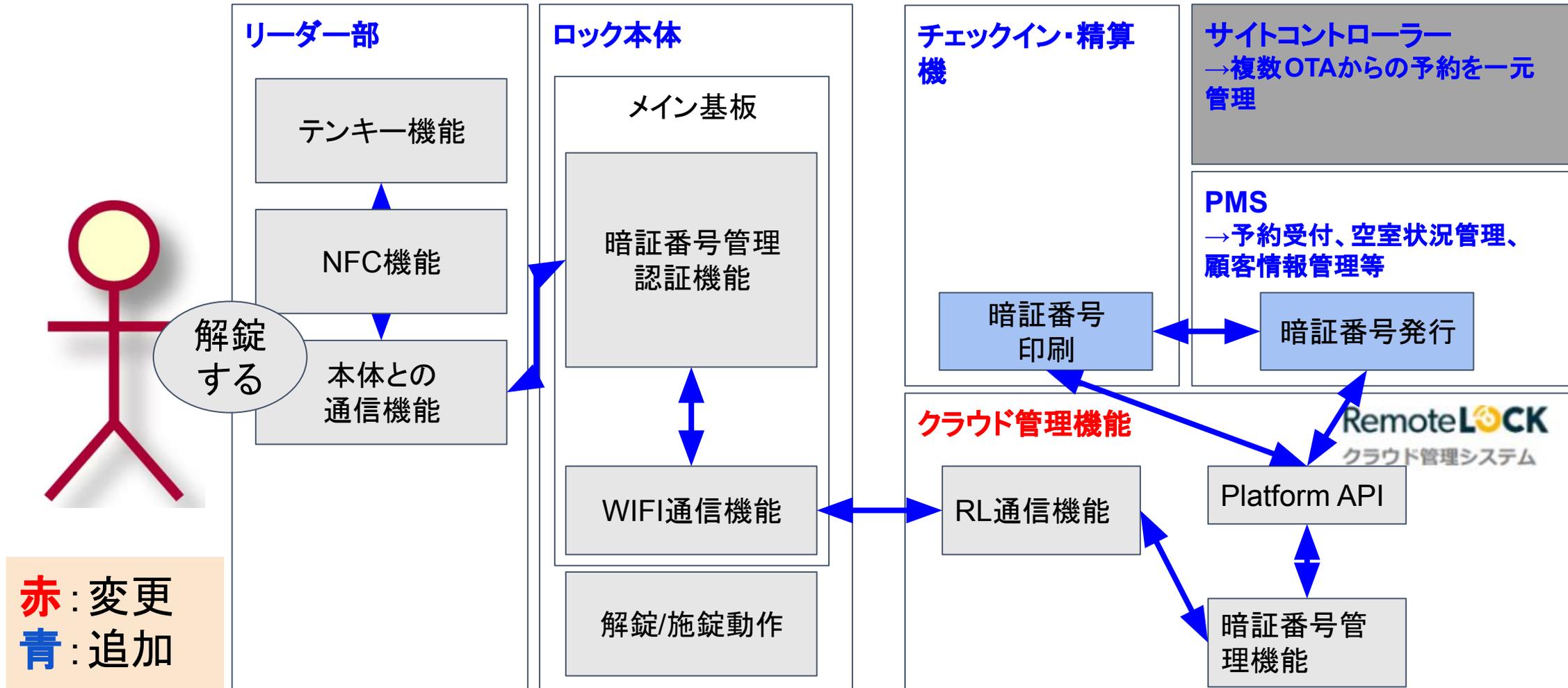
3.3 ホテルシステム連携 (RL導入時)

- 予約情報の流れから見ると、OTA/予約システム→サイトコントローラー→PMS→チェックイン・精算機となる
- RemoteLOCKシステム導入で、チェックイン時にPIN(暗証番号)で解錠が可能になった
→ 経営観点から見ると省人化による固定費圧縮効果大きい



3.4 RemoteLOCK導入時の対応内容

- PMSとチェックインシステム側でRemoteLOCK Platform API呼び出し処理を追加
- OTA/予約システムはブロック図から省略



3.5 対象/問題

対象

- 対象ハードウェアはパートナーで製造。これまでスマートロックではじめての**QRコードリーダを搭載製品の 派生開発**
- **RemoteLOCKクラウドをプラットフォームとして、複数のパートナーのサービスを統合してホテルでのQRコードのユーザエクスペリエンスを実装**

問題

- 営業チームの提案で導入ホテルが決定。2022年8月にホテルの営業開始、6月に現場施工、逆線を引くと2022年3月には製品開発完了が必須！
- 2021年後半にRemoteLOCK 8jのQRコード派生開発は計画していたが15ヶ月かかるとハードウェアパートナーが計画
- **トップ同士で1年前倒し (15ヶ月→3ヶ月)で派生開発を合意**
 - 2022年1月に正式開発スタート
 - **2022年6月設置工事に合うのだろうか？**

3.6 RemoteLOCKとは

RemoteLOCK(リモートロック)とは

- RemoteLOCKはWi-Fiを通じてインターネットに接続し、クラウド上で多数の利用者の入室を管理する
- 遠隔で利用者毎に暗証番号やQRコードなどのカギ発行を行い、ゲストはキーレスで施設の利用ができるようになる

複数の拠点・ドア・入室者を一元管理



細やかな入室制御でセキュリティを向上



運用をより便利にする高度な機能



多彩な機能でビジネスを強かにバックアップ！

クラウドだからメンテナンスフリーで常時最新バージョンの機能をお使いいただけます。

3.7 お客様の課題

経緯: お客様の課題

Before

非効率でムダの多いフロント業務



- 宿泊者カードへの記入
- 物理鍵(カード)の受け渡し、預かり
- フロント行列の整理や誘導
- チェックアウト時の精算対応

After

事務作業を削減しおもてなし強化へ



- 周辺観光の案内
 - ウェルカムドリンクの提供
 - オプションサービスの紹介
- ▶アップセル・クロスセルやファンづくりへ

省人化で人手不足に対応。人は付加価値の高い業務に注力

3.8 お客様の要求

経緯: お客様の要求

暗証番号やQRコードが鍵代わりに

- ✓ 対面での鍵のやり取りが不要
- ✓ キーレス化で自由な滞在へ

非対面
非接触

キーレ
ス化

オンラ
イン化



店舗業務の無人化と
体験価値の向上へ

QRコード解錠イメージ動画

施設全体を制御!

①入口自動ドア②エレベータ③客室



3.9 お客様の要求

経緯:ステークホルダー

- WEST inx社
 - RemoteLOCK 8j リーダー側スキャナー搭載
- RL社メイン基板担当:QRコード処理追加
 - RemoteLOCK 8j インナーユニット内のPCB
- RL社クラウド開発担当:QRコード処理追加
 - RemoteLOCKクラウド管理画面
- PMS/チェックインシステム担当社
 - QRコード処理追加

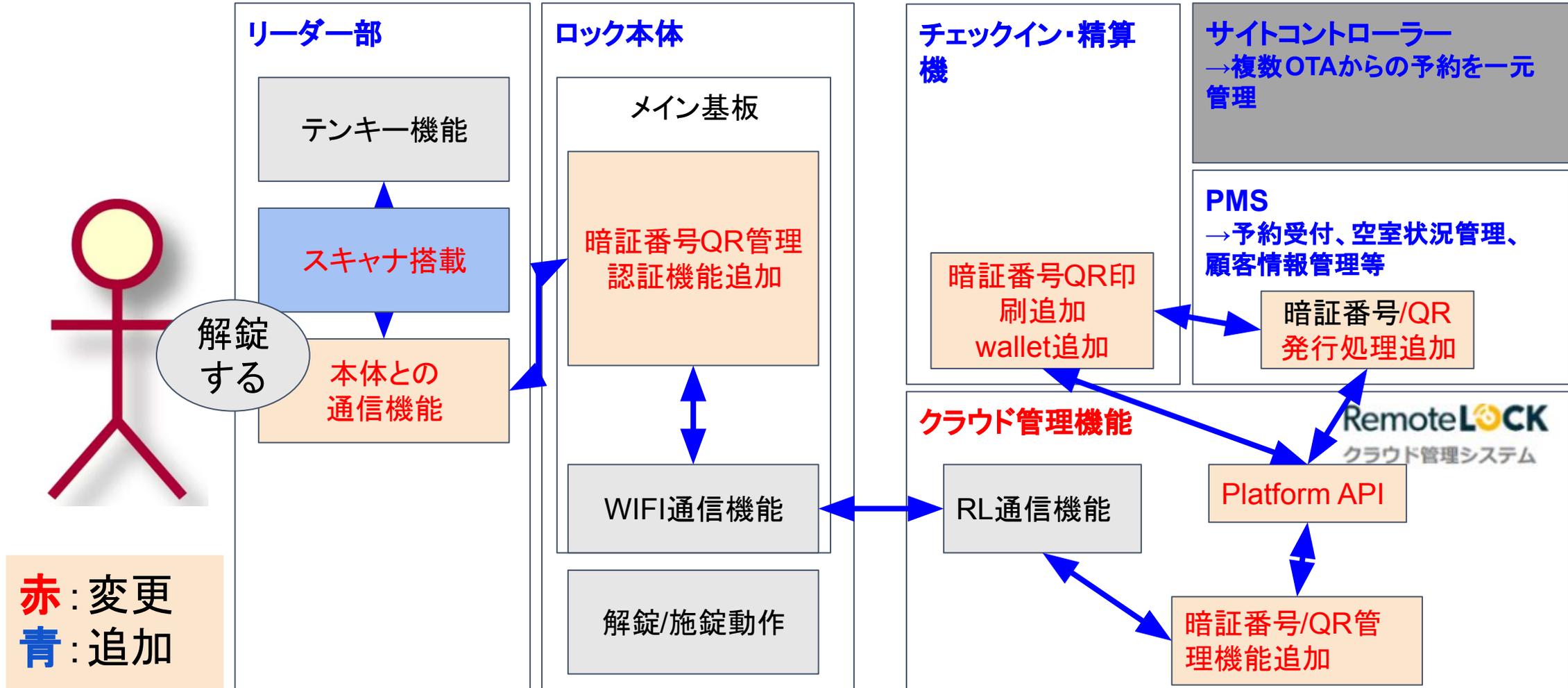
※施工については以下と調整が発生

- 施主:霞ヶ関キャピタル社 (FAV HOTELシリーズを全国15拠点で展開、全てRemoteLOCK)
- 施工:ゼネコン、サブコン、電設会社、建具設備会社、鍵屋



3.10 QRコード導入時の対応内容

変更箇所は以下の通り。OTA/予約システムはブロック図から省略



3.11 解決: USDM(eXDDP)

- RemoteLOCK8j-Q派生開発についてUSDMを記載、ステークホルダ(パートナー)を列に並べる。段階的に詳細化▶詳細は各社で検討。USDMからパートナーの詳細設計にリンク
- USDMの記載内容の変遷
 - システム間IF変更も含め列挙。パートナー列は影響のあり、なしを記載
 - 該当システム/サブシステムの修正内容について記載
 - パートナーと意識合わせ 修正を繰り返す

#	種別	内容	対応時期	Where							ALMEX社	施工影響	その他
				RemoteLOCKクラウド	メイン基板	WEST-inxデンキキーリーター(基板群)	センサー基板	モーター基板	金型・部品				
6.1	要求	メイン基板・クラウド側でデンキキーリーダが、FelIcaがQRコード対応か識別可能にする。	QR-A1										
6.2	理由	外部API経由でのPIN発行時に、QRコードがIdmにするか決定できるようにするため。											
6.3.1	仕様	デバイス設定画面に「リーダ種別」メニューを追加し、FelIca、QRコードの2種類とする。 ・デフォルトはFelIcaとする。		あり	なし	なし	なし	なし	なし	なし	なし	なし	
		上位4bitは0xF、下位4bitは送信バイトを示す。 上位4bitは0xE、下位4bitは送信バイトを示す。 は、PINコードのフィールドではなく、ICカード(FelIca)と同様に、TOBIRAと共通で管理・認証するという仕様とあわせるため。		あり	あり	あり	なし	なし	なし	なし	なし	なし	
		8bytesのデータとする。4 - 最大16桁(0-f) 0xFFFFFFFFFFFFFFFFのように0xを付与したQRコードを作る。 1234567890 でエンコードされたQRコードを読み取り可能にする。 は、数字で桁数チェックなしで進めたい。		あり	あり	あり	なし	なし	なし	なし	なし	なし	
		すき、想定利用環境での適正な読み取りレベルの担保するた	QR-A1										
		【(パネルデザイン) QRコードロックはブラックパネルとする。		なし	なし	なし	なし	なし	あり	なし	なし		
6.3.2	仕様	【(パネルデザイン) 下記は一案。利用者へのわかりやすさを優先したい。ブラックパネルがNGの場合はシール同梱なども想定したい。 カメラへの距離をわかるようにしてほしい。 < QR提示位置のわかりやすさ1> カメラ周囲にリンク状の発光部分を設けてQRコードをかざす場所を示す。 < QR提示位置のわかりやすさ2> さらに(パネル印字で、カメラ周囲にQRコードを模した白のアイコンを入れる。(リンクアイコンの外周または横)		なし	なし	なし	なし	なし	あり	なし	なし		
		<ロゴ> ロゴは、派生開発推進協議会、WESTとRemoteLOCKロゴを併用する。											

ビジョン
制約条件
IF変更
etc...



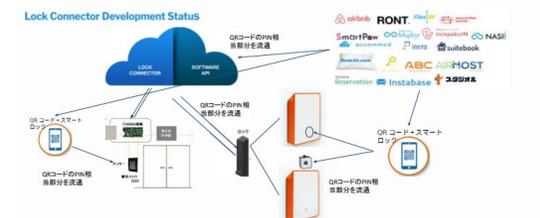
How
は各社内部の活動

3.12 解決:パートナーとのコミュニケーション

- マルチパートナーでのプロジェクト管理・特にWESTinx社とは週1で検討会を開催
 - 影響範囲の識別・明確化
 - システム制約事項
 - 修正箇所の最小化
 - 機能と期間のトレードオフ
 - ロードマップをパートナー間で合意
 - 開発と製造スケジュール短縮
- IoTデバイスの製造と設置で3ヶ月
 - ゼネコン・サブコンと調整

4. HWの変更イメージ

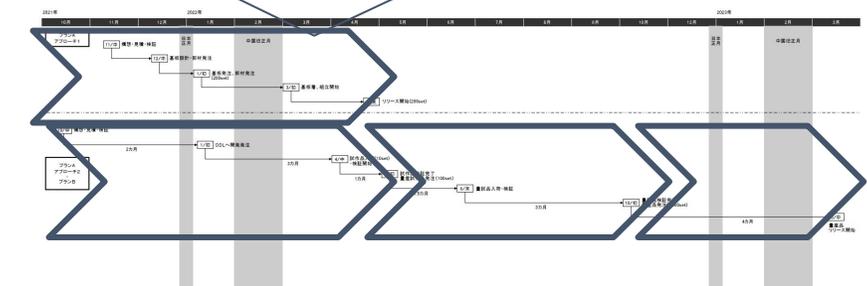
E06のFelCa対応をベースにしてQRコード対応について検討した。HWの変更イメージは下記2つ。
 案1. 既存テンキーリーダーのNFCリーダ部分に取り付ける
 -FelCaは読み取りできなくなる。耐雑性は担保しやすい。設置性も担保しやすい。
 案2. テンキーリーダー上部(もしくは別の場所に)追加で取り付ける
 -FelCaと並列で扱える=ACS(TOBIRA)リーダと同様。耐雑性は劣る可能性がある。



QRコードに設定する内容としては下記が想定される。それぞれメリット/デメリット(Pros/Cons)を一覧にまとめる

#	方式	メリット(Pros)	デメリット(Cons)
1	4-10桁の数値として扱う(問題があれば変更する)	<ul style="list-style-type: none"> ● ACS(TOBIRA)のFelCa対応と仕様が共通化可能 ● クラウド変更不要 ● メイン基板変更不要 	<ul style="list-style-type: none"> ● 文字列は扱えない ● RLクラウドもしくはラッパーAPIでQRコード作成とメール添付処理が発生
2	外部システムの利用を前提に、外から与えられたASCII文字列からPINコードをテンキーリーダーで導出して認証	<ul style="list-style-type: none"> ● 人間が認識しやすくなる 	<ul style="list-style-type: none"> ● テンキーリーダーの処理があるため、認証までのタイムラグが発生する可能性がある。 ● TOBIRAのQRコードルールと別になっってしまう。
<p>★前提: QRコードのPIN部分のみ読み取って認証する方式。 例えば、xxxxxxxxxxxx(文字列、URLなど)-12345678の12345678がPIN部分とする考え方</p>			
3	外部システムの利用を前提に、外から与えられたASCII文字列を流過して認証	<ul style="list-style-type: none"> ● 人間が認識しやすくなる 	<ul style="list-style-type: none"> ● RLクラウド、メイン基板、テンキーリーダーがすべて変更が発生する。 ● メイン基板のメモリを圧迫し、記憶できるコード数が減少する。

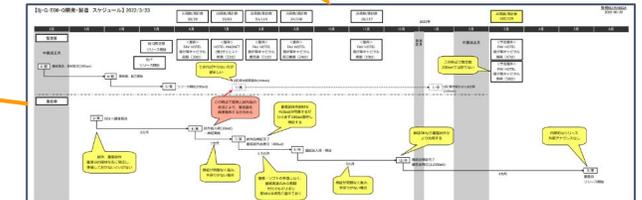
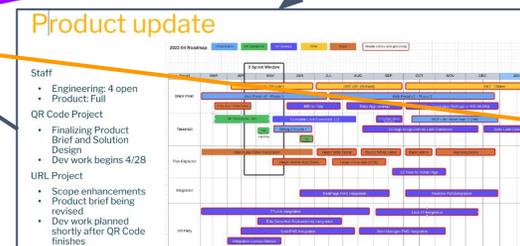
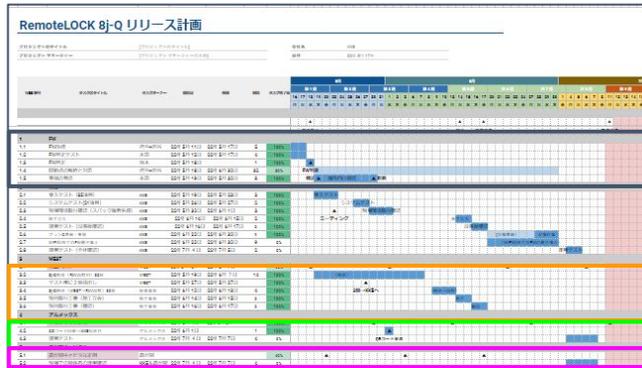
**USDMをアンカーとした
パートナーとのコミュニケーション、
合意形成の短縮化**



3.13 解決: TMによるトレーサビリティとスケジュール管理

- USDMがアンカーになる
- パートナーのスケジュールをベースにテスト計画をKKEで作成・実施
- 施工のための資料も作成
- 2022年6月までに施工完了

#	種別	内容	対応時期	RemoteLOCK		VEST linx		ALMEX社	施工影響	その他
				クラウド	メイン基板	デンソーリーダ(基板群)	センサー基板			
6.1	要求	メイン基板・クラウド側でデンソーリーダが、FelCaかQRコード対応が選択可能にする。	QR-A1							
6.2	理由	外部API経由でのPIN発行時に、QRコードがIdmiにするか決定できるようにするため。								
6.3.1	仕様	デバイス設定画面に「リーダ種別」メニューを追加し、FelCa、QRコードの2種類とする。 ・デフォルトはFelCaとする。		あり	なし	なし	なし	なし	なし	なし
6.3.2	仕様	[2111012追加] QRコードの場合、上位4bitは 0xF、下位4bitは送信バイトを示す。 FelCaの場合、上位4bitは0x、下位4bitは送信バイトを示す。 理由: QRコードは、PINコードのフィールドではなく、ICカード(FelCa)と同じ領域でクラウド・TOBIRAと共通で管理・認証するという仕様とあわせるため。		あり	あり	あり	なし	なし	なし	なし
6.3.3	仕様	[2111130追加] FelCa Idmと同じ8bytesのデータとする。4 - 最大16桁(0 - f) ・16進の場合は、0xFFFFFFFFFFFFFFFFのように0xを付与したQRコードを読み取り可能とする。 ・10進の場合は、1234567890 でエンコードされたQRコードを読み取り可能とする。 ・デンソーリーダは、数字で桁数チェックなしで進めたい。		あり	あり	あり	なし	なし	なし	なし
7.1	要求	【デザイン、カメラ】	QR-A1							
7.2	理由	QRコードのわかりやすさ、想定利用環境での適正な読み取りレベルの担保のため								
7.3.1	仕様	【(ネルデザイン)】QRコードロックはブラック(ネル)とする。		なし	なし	なし	なし	あり	なし	なし
7.3.2	仕様	【(ネルデザイン)】下記は一覧、利用者へのわかりやすさを優先したい。ブラック(ネル)の場合にはネル特長もも考慮したい。カメラへの邪魔をわかるようにしてほしい。 < QR提示位置のわかりやすさ1 > カメラ周囲にリンク状の発光部分を設けてQRコードをかざす場所を示す。 < QR提示位置のわかりやすさ2 > さらに(ネル印字で、カメラ周囲にQRコードを模した白のアイコンを入れる(リンクアイコンの外周または横) < ロゴ > ロゴ3.5倍拡大の場合、VESTとRemoteLOCKのロゴで表現してほしい。		なし	なし	なし	なし	あり	なし	なし



課題説明とグループワーク

4. 課題説明

1. シチュエーション説明
2. ブロック図説明

5. グループワークと発表

1. USDM作成
2. TM作成→影響個所の特定
3. セルを埋める→変更仕様(どこでどこのロジック、処理を行うかを明確にする)
4. チーム結果共有

6. 解説・まとめとQ&A

- a. 解説
- b. まとめとQ&A

4.1 シチュエーション説明

● ライブラリ

- QRコード拡張時にテンキーリーダーにQR用カメラを追加済み。このモジュールは認識した顔情報の特徴点を抽出し数値情報にハッシュするライブラリの機能追加で顔認証機能が追加できる

● チェックイン端末

- 顔を撮影して記録する機能が存在する。これに上記と同じ顔認証ライブラリを追加する。ハッシュされた特徴点情報による数値情報を暗証情報とする。この暗証情報をチェックイン時にRemoteLOCKクラウド管理サービスに通知する
- QRコードと顔認証のいずれかを使うかの選択機能を追加する

● RemoteLOCKクラウド管理サービス

- QRコードと顔認証のいずれを使うかの選択情報を追加する
- RemoteLOCK本体に暗証情報を通知する

● RemoteLOCKのテンキー

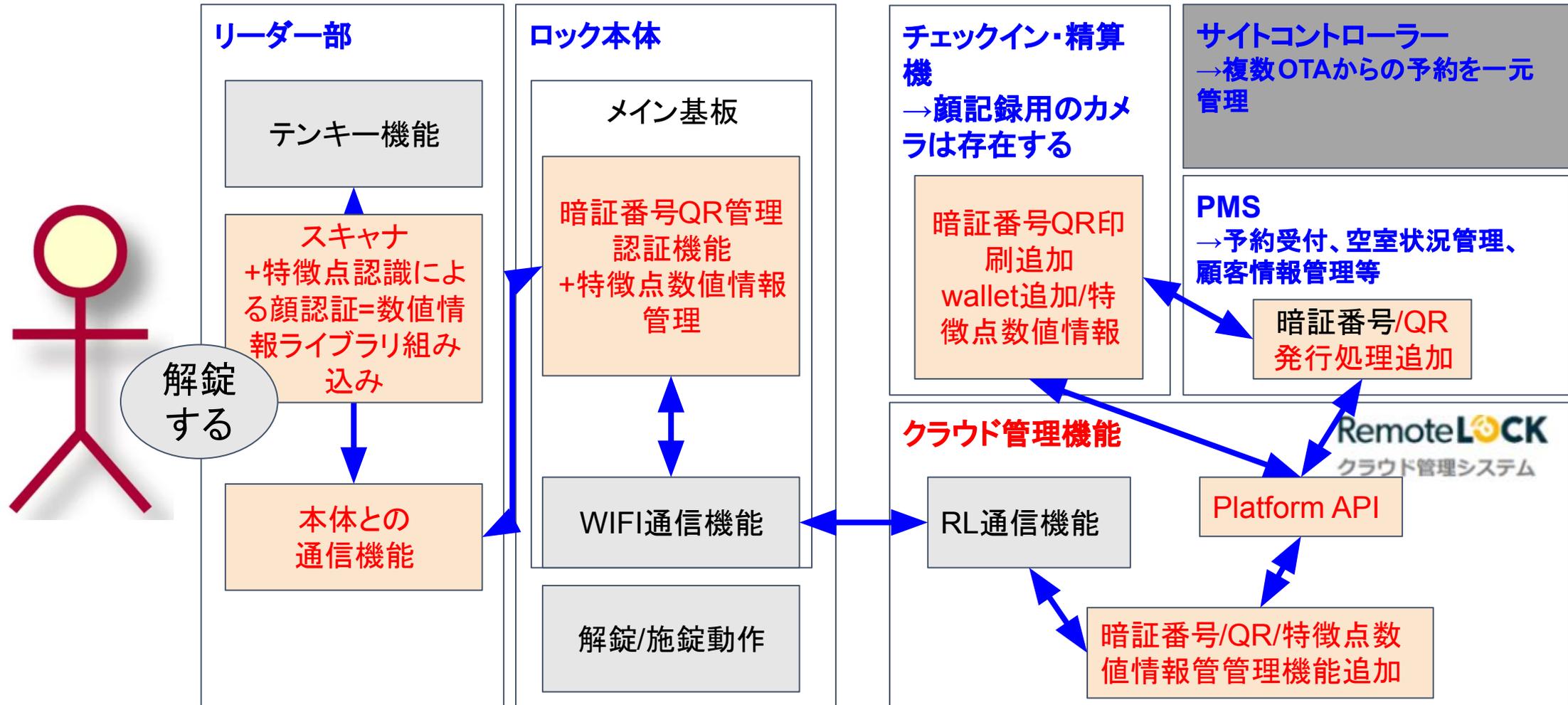
- 顔認証の特徴点情報による数値情報本体に通知する

● RemoteLOCKロック本体のメイン基板

- 顔認証の特典情報による数値情報で認証する。認証OKの場合は解錠する

4.2 顔認証導入時の対応内容 (課題)

- 変更箇所は以下の通り。+特徴点認識による顔認証=数値情報ライブラリを組み込むことで対応
- チェックイン装置にはカメラは存在する



グループワークと発表

1. USDM作成
2. TM作成→影響個所の特定
3. セルを埋める→変更仕様
(どこでどこのロジック、処理を行うかを明確にする)
4. チーム結果共有

解説・まとめと Q&A



派生開発推進協議会

AFFORDD