UNITIKA
We Realize It!

JUASスクエア２０２０
オンラインペアセッション

ユニチカ事例から見るITトレンドの選別・導入と業務革新　－システムリフォームの進化

ユニチカ株式会社
管理本部 情報システム部
2020年7月29日

Contents

会社概要

ユニチカ株式会社

創立	1889年（明治22年）6月19日
資本金	100,450,000円
主な事業内容	•高分子事業 フィルム（ナイロン・ポリエステル） 樹脂（ナイロン・ポリエステル・ポリアリレート） •機能材事業 機能材（ガラス繊維、ICクロス、 ガラスビーズ、活性炭繊維） 不織布 （ポリエステルスパンボンド、綿スパンレース） 生分解材料 •繊維事業 産業資材 衣料・生活雑貨・寝装 バイオマスプラスティック（テラマック）

UNITIKA LTD.

プロジェクト概要
（基幹システム・リプレース）

プロジェクト概要

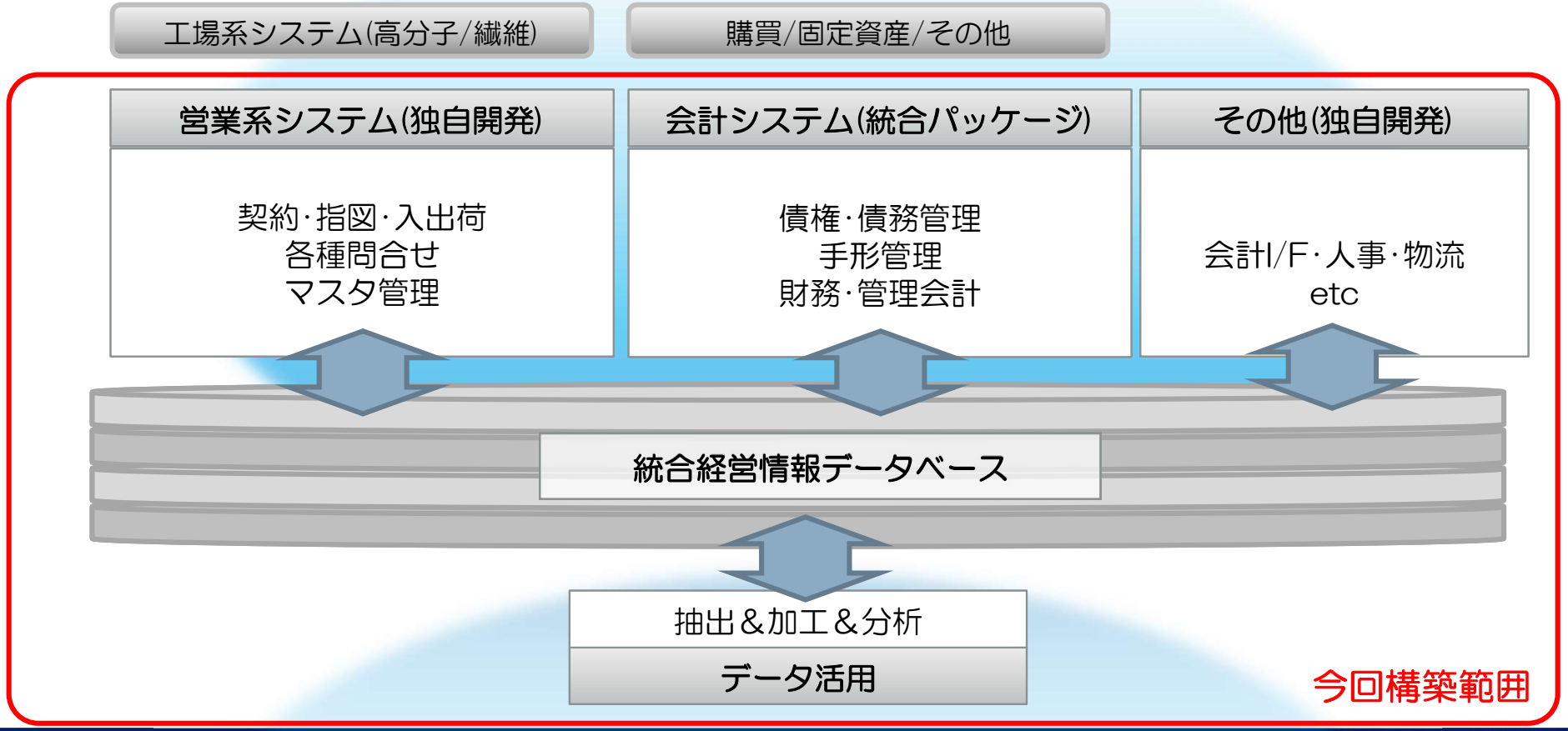
プロジェクト背景　～現行ホスト利用継続の場合の問題～

- **ホスト機の保守切れ（2021年3月）**
  - バージョンアップに必要な作業期間は、6年
  - バージョンアップにかかる費用は、数億円規模

- **バージョンアップを完了したとしても**
  - 古くてマイナーな言語（PL/1）の保守要員が将来的に確保困難。
  - ホストのランニング費用は高額のまま。
  - 監査法人より、監査、統制面で品質向上の要望がきている。
  - 経営判断に要するデータの収集、加工についてスピード、粒度について課題多数。
  - このままでは、他システム連携や取引先との密な連携、グローバル対応などが困難。

プロジェクト概要

プロジェクト概要　～対象システムの構成～



プロジェクトの歩み

2011年3月～2012年12月

SWINGプロジェクト

SOAによる基幹システムの再構築

※パイロット的な位置付け

2013年10月～2017年10月

COMPASSプロジェクト

システムリフォームにより、
ホストシステムと同じ機能を
実現

ユニチカ
＋
大手SIer

ユニチカ
＋
SoftRoad

- そもそも基幹システムの再構築は困難
- コスト、開発期間の制約

方向転換の理由

- そもそも基幹システムの再構築は困難
  - **規模が大きい**（画面数1400以上、JOB数2200以上）
  - **仕様書が無い**
  - 基幹システムが多くの他システムと連携している

- コスト、開発期間の制約
  - メインフレームの保守切れのタイムリミット
  - 今の業務を急に変更することが困難
  - **今の仕様と合致していることを確認できる有識者が不足**

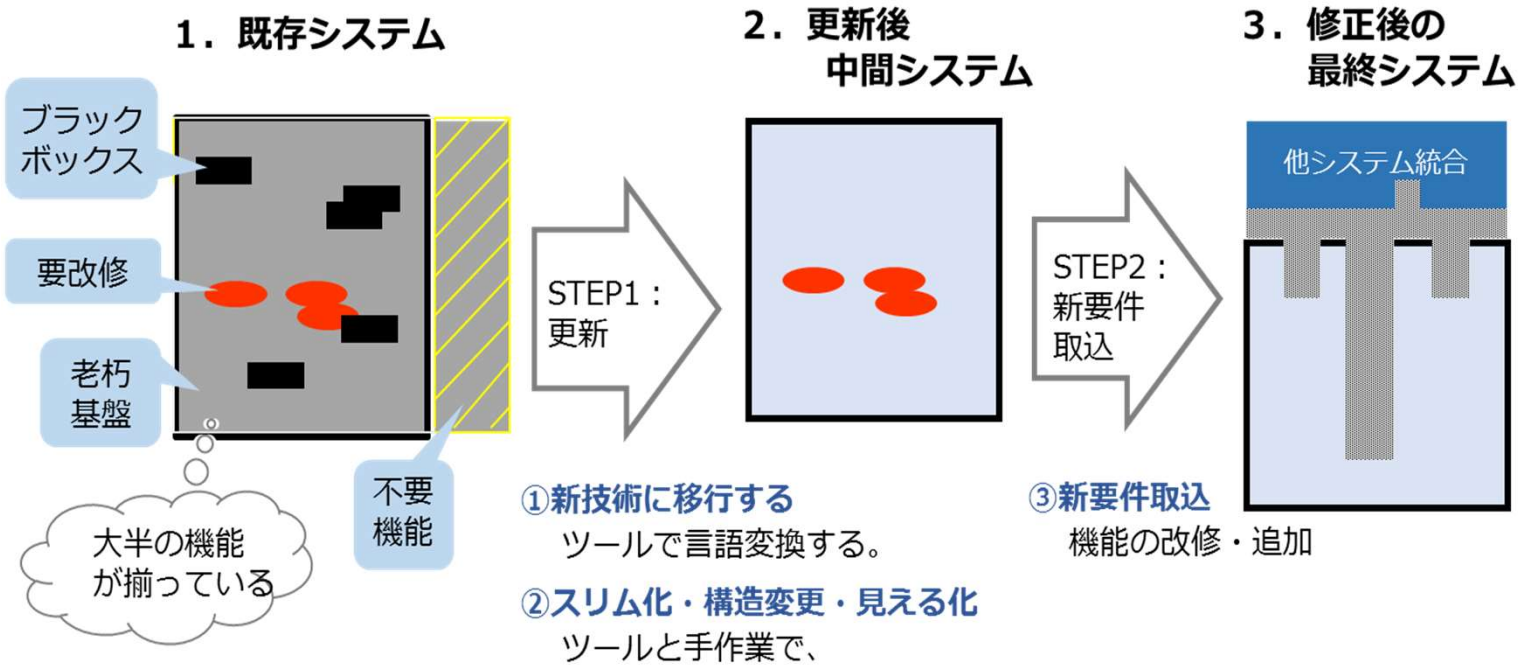
これらの問題をシステムリフォームが解決してくれた。

システムリフォーム・ユニチカモデルとは

システムリフォーム・ユニチカモデルとは

システムリフォームとは

既存のシステムリソースをいかしながら、システムを最新のIT基盤にリフォーム（引越し）する手法です。



▪ メリット

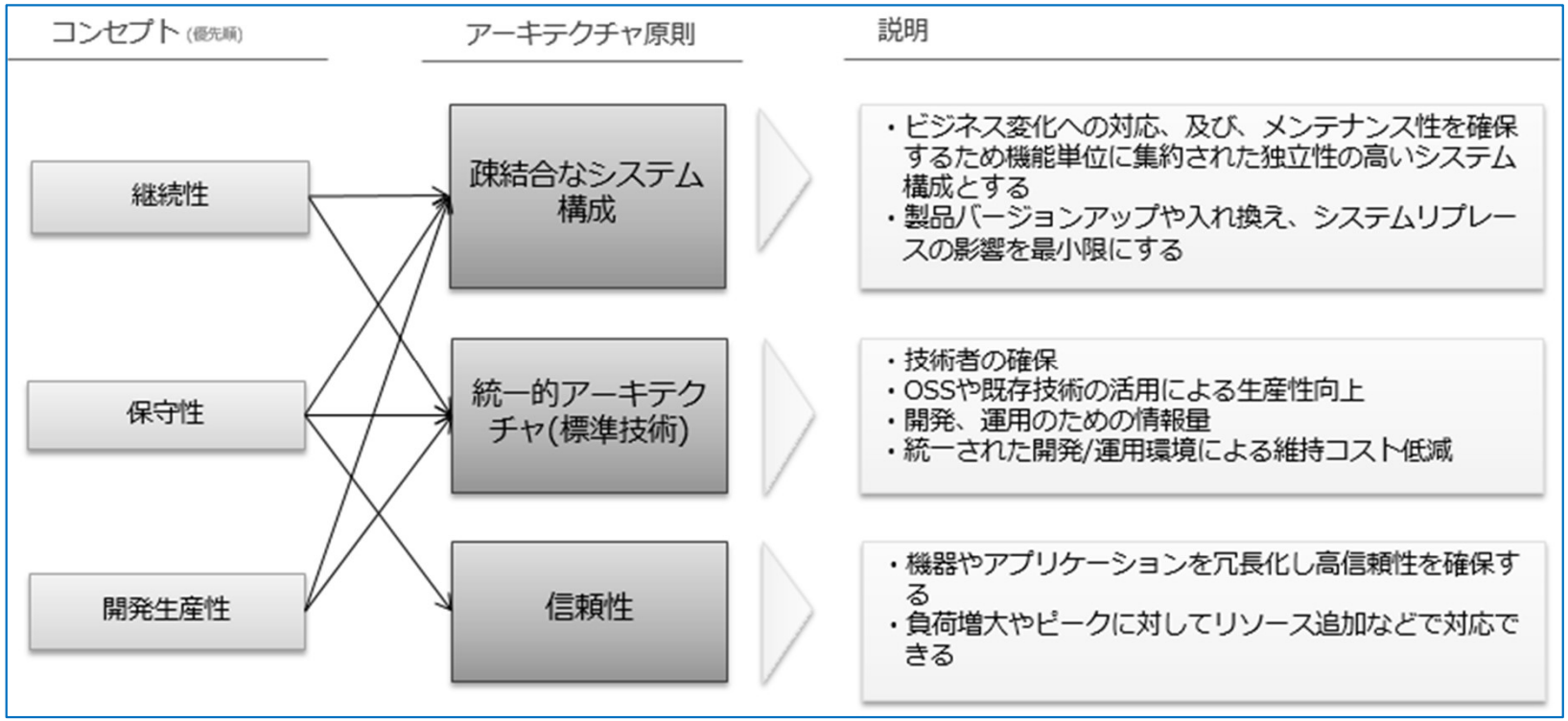
- 開発コストが抑えられる。
- 通常のスクラッチ開発と比較して品質が高い。
- 発注側（ユニチカ側）の作業負荷（調査工数 等）が軽減される。

システムリフォーム・ユニチカモデルとは

ユニチカモデルとは　～アーキテクチャ（原則）～

- アプリケーション基盤の構築にあたり、

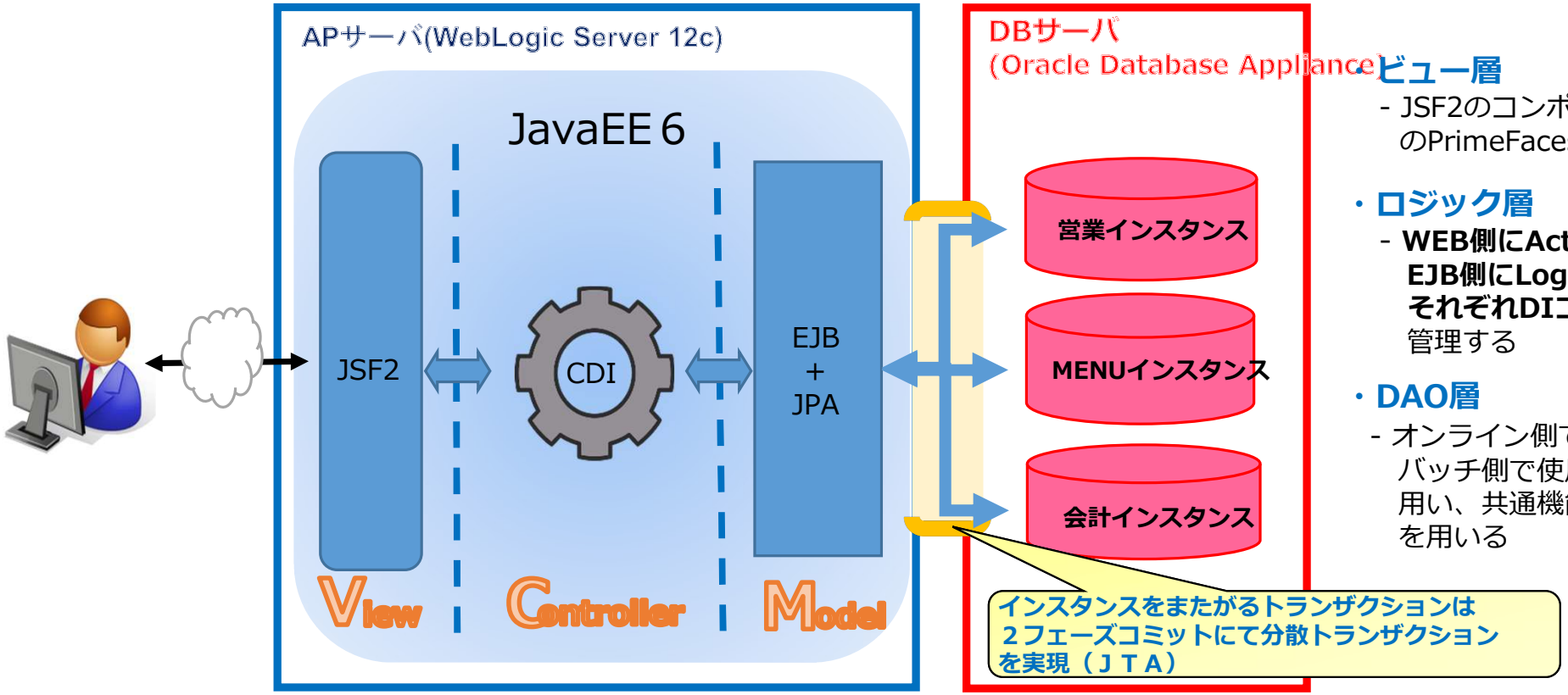
3つのコンセプトを元にアーキテクチャ原則を定めた。



システムリフォーム・ユニチカモデルとは

ユニチカモデルとは　～アーキテクチャ編（１）～

Compassプロジェクト共通で利用するWebフレームワークを作成。
JavaEE6のフルサポート及び代表的なOSSに拘った製品群でアーキテクチャを構成している。



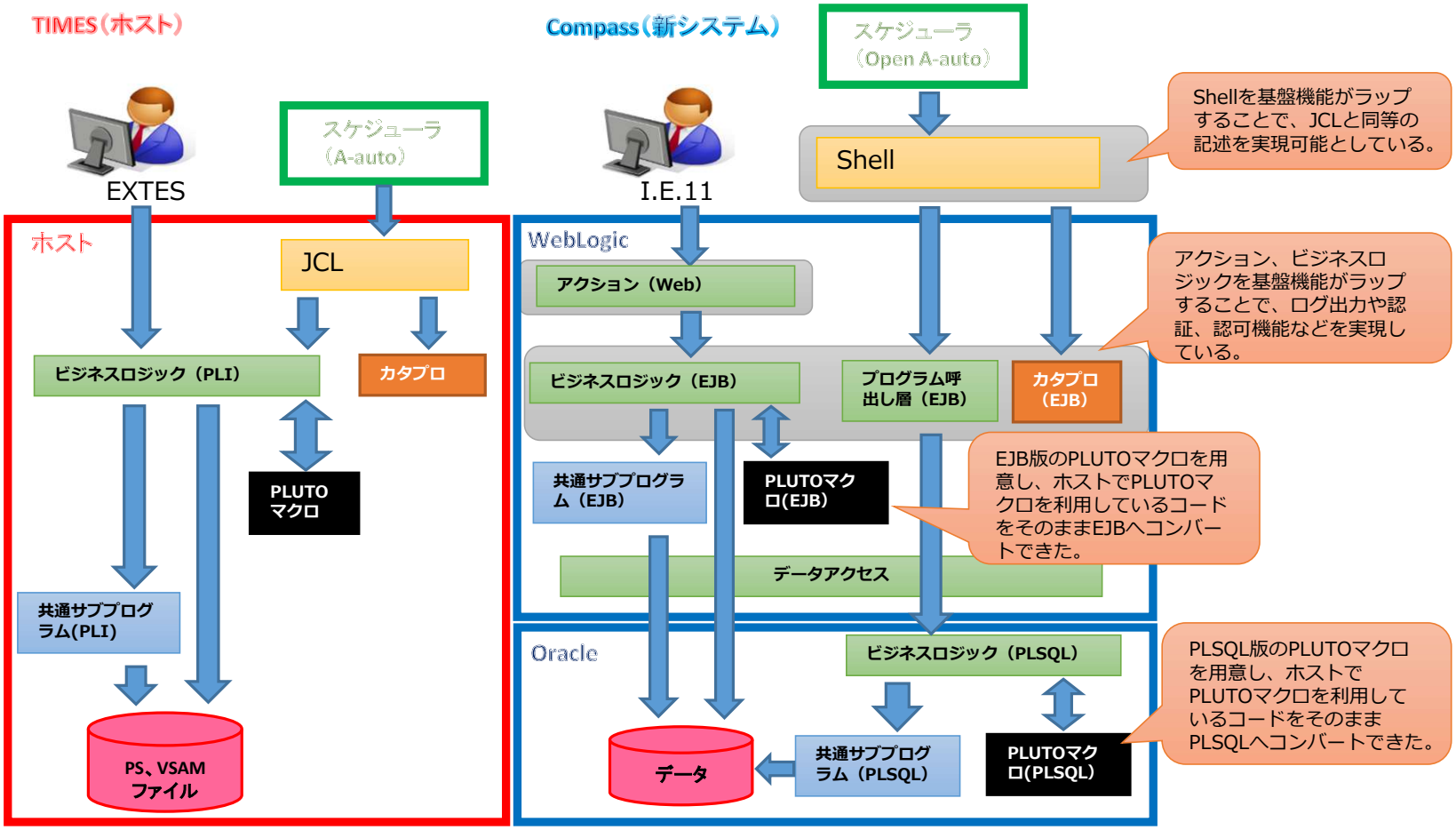
・ビュー層
- JSF2のコンポーネントライブラリのPrimeFacesを使用

・ロジック層
- **WEB側にActionクラス**を配置し、**EJB側にLogicクラス**を配置。**それぞれDIコンテナ（CDI）**で管理する

・DAO層
- オンライン側で使用する場合はJPAを、バッチ側で使用する場合はJDBCを用い、共通機能については全てJPAを用いる

システムリフォーム・ユニチカモデルとは

ユニチカモデルとは　～アーキテクチャ編（２）～



- DIコンテナによりロジック層とビュー層、DAO層を分離し疎結合を担保。
- 更に、アプリ基盤でサンプル画面やテンプレートを用意し、できる限り疎結合の関係を保てるように工夫
- テンプレート例
  - ・プロパティクラス
  - ・Actionクラス
  - ・DTO&Logicクラス
  - ・画面XHTML
- サンプル例
  - ・メニュー登録
  - ・グループメニュー登録
  - ・パターン別入力画面
- オンライン開発実例等も用意
  120頁超の詳細なサンプル説明資料も
  ・オンライン編
  ・バッチ編
  ・ロジック編
  に分けて提供

システムリフォーム・ユニチカモデルとは

ユニチカモデルとは　～データベースデザイン編（１）～

レガシーファイル→RDBMS(Oracle)の変換を実現。これをシステムリフォームに適用可能とした、ルールの一部を紹介します。

- **配列要素 > 第1正規化**
  - 要素数が4以上の配列は、子テーブル化。

＜イメージ＞

現行形式

APRMSTR(契約ﾏｽﾀ)

等級1～3	単価1～3	・・・	色柄明細1～10

配列パターン（第一正規化）

1	KYC#FD	10	712		
2	C#		712	5	CHA
2	C#KYQ		717	72	PAC
2	C#SZQ		721	72	PAC
2	C#FNSQ		725	72	PAC
2	C#CNSQ		729	72	PAC
1	KYC#FD	10	712	21	CHA
1	KYSKNA		922	25	CHA
1	KYSENA		947	25	CHA
1	NONAME		972	22	CHA

新テーブル

APT_MSTR(契約ﾏｽﾀ)　主ﾃｰﾌﾞﾙ

主ｷｰ	等級1～3	単価1～3	・・・

＋

APT_MSTR_KYCNOFD(契約ﾏｽﾀ色柄明細)　ﾃｰﾌﾞﾙ

主ｷｰ	明細No	色柄明細情報

等級・単価など3以内のものは主ﾃｰﾌﾞﾙにそのまま横持ちで格納

色柄明細、4以上のものは明細ﾃｰﾌﾞﾙに分割、縦持ちで格納

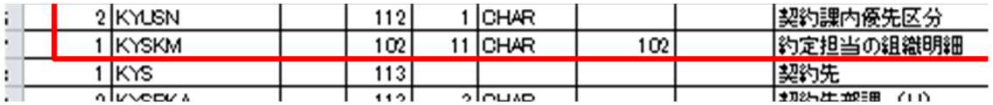
（注意事項）

① 基本的には、上記のルールだが、ﾕﾆﾁｶ希望により（将来を見越してor処理簡素化）、3つ以内だが、明細ﾃｰﾌﾞﾙに持たせたい場合、4つ以上だが、明細に分けたくない場合が出てくるので、設計後、一旦ﾕﾆﾁｶのレビュー確認後、開発作業に入るようにしていただく。

システムリフォーム・ユニチカモデルとは

ユニチカモデルとは　～データベースデザイン編（２）～

- **複合コード > 最小単位ルール**
  - 最も小さい単位で、1項目とする。



2	KYUSN		112	1	CHAR		契約課内優先区分
1	KYSKM		102	11	CHAR	102	約定担当の組織明細
1	KYS		113				契約先

＜イメージ＞

現行形式

組織明細		
会社	部門	・・・

新ﾃｰﾌﾞﾙ

会社	部門	・・・

最小単位でDB項目とする。

※）同種の例
契約No、指図No、入出荷No。　これらは現行、下記の構成となっているが、それぞれ新ﾃｰﾌﾞﾙとしては最小単位で項目を持つ

＜イメージ＞

入出荷No			
指図No			枝3
契約No		枝2	枝3
発連	枝1	枝2	枝3

新ﾃｰﾌﾞﾙ

発連	枝1	枝2	枝3

発生連番

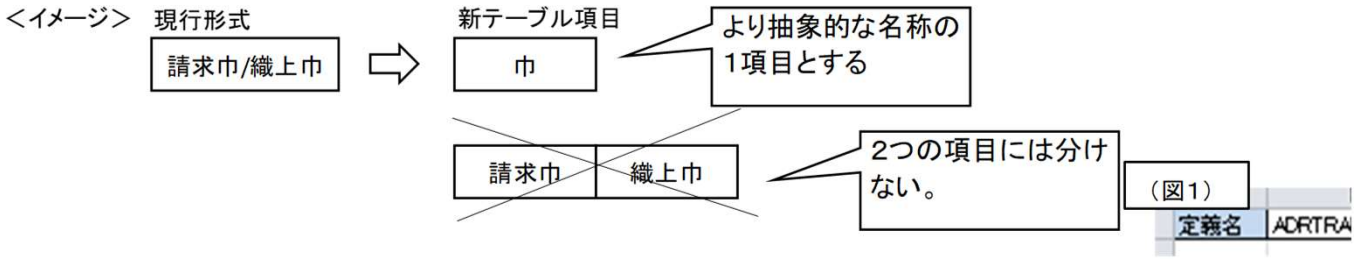
但し、現行システムで、例えば"入出荷No"として項目にｱｸｾｽしている箇所があれば、
JavaのEntityｵﾌﾞｼﾞｪｸﾄに、入出荷Noとしてｱｸｾｽするgetter/setterを定義し
"入出荷No"として扱えるように新システムをコーディングするものとする。

（例外事項）
① 下記の項目については、最小単位で設定しない。
A）最終更新日時・・最小単位ルールに従うと、「最終更新年月日」「時」「分」に分かれてしまうが、そうせずDB設計規約に従った最終更新日時項目に置き換わる。※）決済条件は例外でなく
B）物的属性・・・最小単位ルールとすると「物的属性1桁目」「2桁目」に分かれるがそうせず、新ﾃｰﾌﾞﾙでは、「物的属性」(2桁)の1項目として定義する。　項目を分けてDB化する

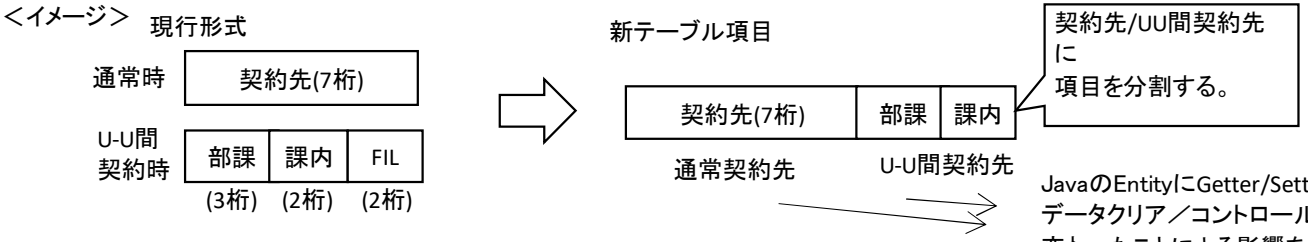
システムリフォーム・ユニチカモデルとは

ユニチカモデルとは　～データベースデザイン編（3）～

- **1項目複数意味項目（同一タイプ・レイアウト） > 項目名の抽象化**



- **1項目複数意味項目（別項目格納） > 項目分割**
  - 別項目として定義する。



JavaのEntityにGetter/Setterでの工夫はしない。
データクリア／コントロールブレーク処理等の処理では、持ち方が
変わったことによる影響を考慮した新システムのﾌﾟﾛｸﾞﾗﾑとすること

システムリフォーム・ユニチカモデルとは

ユニチカモデルとは　～データベースデザイン編（4）～

- **マルチレイアウト > テーブル分割**

＜イメージ＞

現行形式

APRMSTR(契約ﾏｽﾀ)

主キー	共通部分	識別区分	レイアウト01/レイアウト02

マルチレイアウト部分

4	2	GGNTNT		616
5	2	GTEHM		617
6	2	GSHQRU		619
7	2	SKTNQR		626
8	2	GSKKTT		633
9	2	DBTKZK		634
0	2	DSHN		636
1	2	DTFN		640

⇨

新テーブル

APT_MSTR(契約ﾏｽﾀ)　主ﾃｰﾌﾞﾙ

主キー	共通部分	識別区分

＋

APT_MSTR_APM01　子ﾃｰﾌﾞﾙ(レイアウトAPM01)

主キー	レイアウト01

APT_MSTR_APM02　子ﾃｰﾌﾞﾙ(レイアウトAPM02)

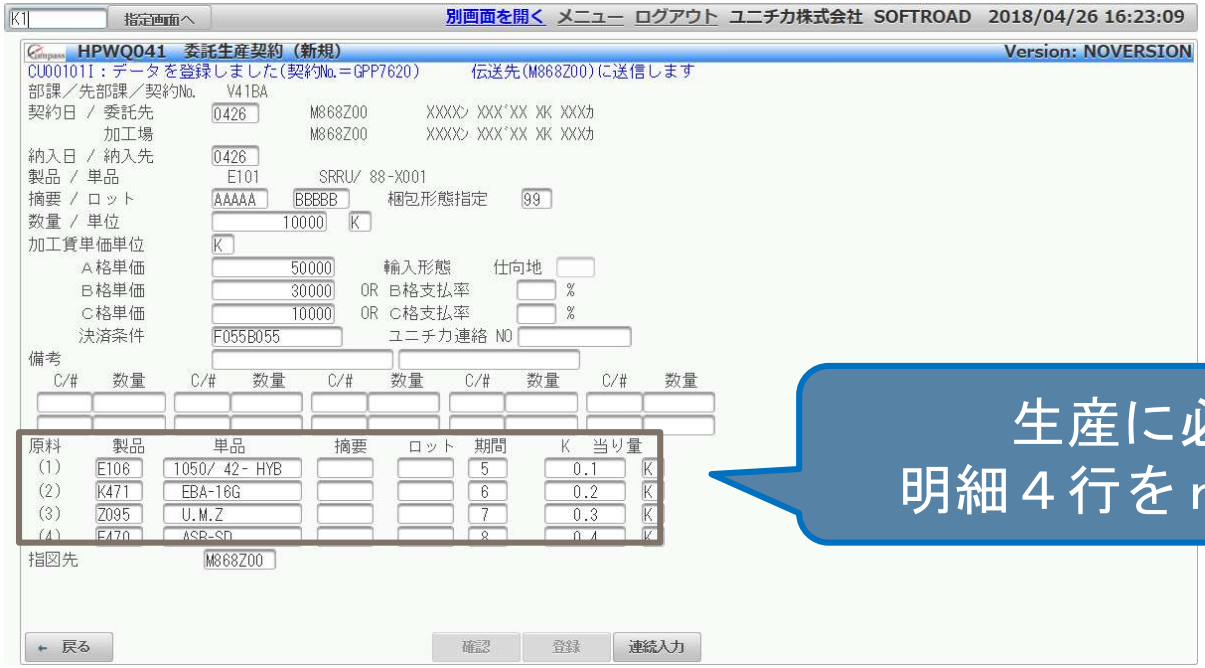
主キー	レイアウト02

- **前述の基本ルール決定に加えて、新テーブルレイアウト設計については、全レイアウトをレビュー**
  - 例外項目、例外テーブルについては、レビュー指摘により妥当に訂正
  - レビュー確定後の新テーブルレイアウトを加味した、移行開発を実施

システムリフォーム（ユニチカモデル）であればこそ容易に実現できた事例①：DB正規化

- 樹脂の原料個数の拡張（４レコード → ｎレコード）

テーブルを正規化（子テーブル化）し、入力項目数の拡張が容易となった。

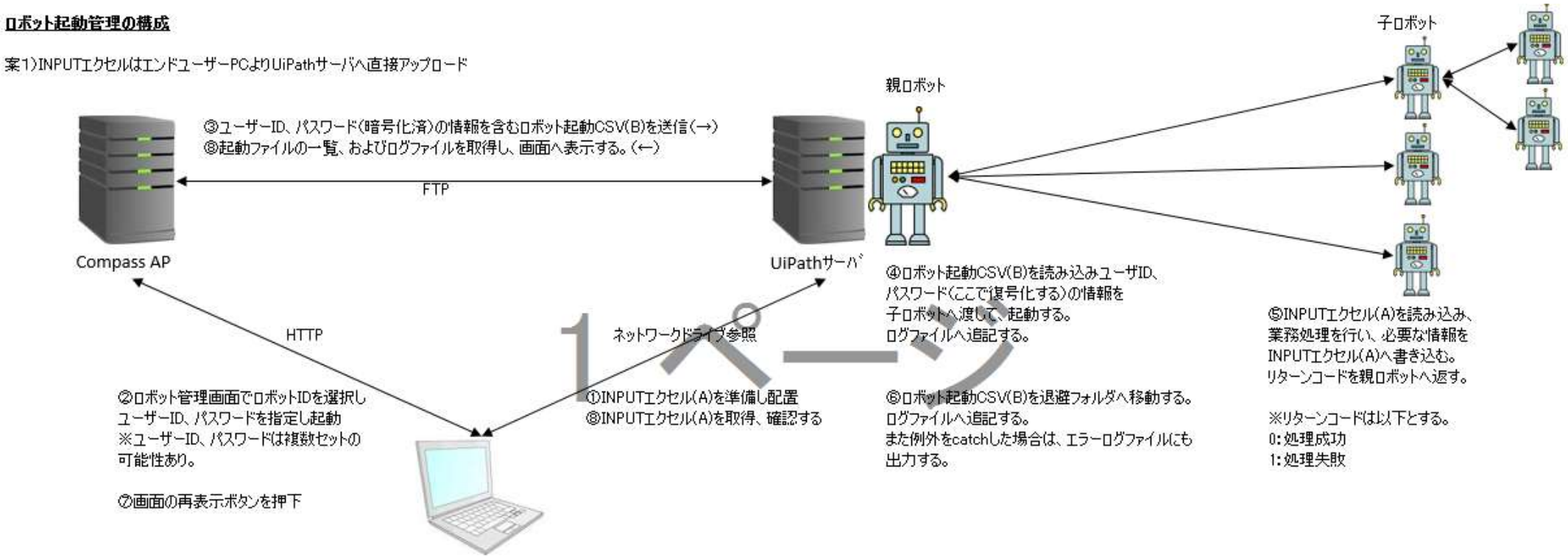


生産に必要な原料
明細４行をｎ行に拡張する

システムリフォーム（ユニチカモデル）であればこそ容易に実現できた事例②：外部アクセス機能

- **ＲＰＡ**（Robotic Process Automation）ロボットの管理機能

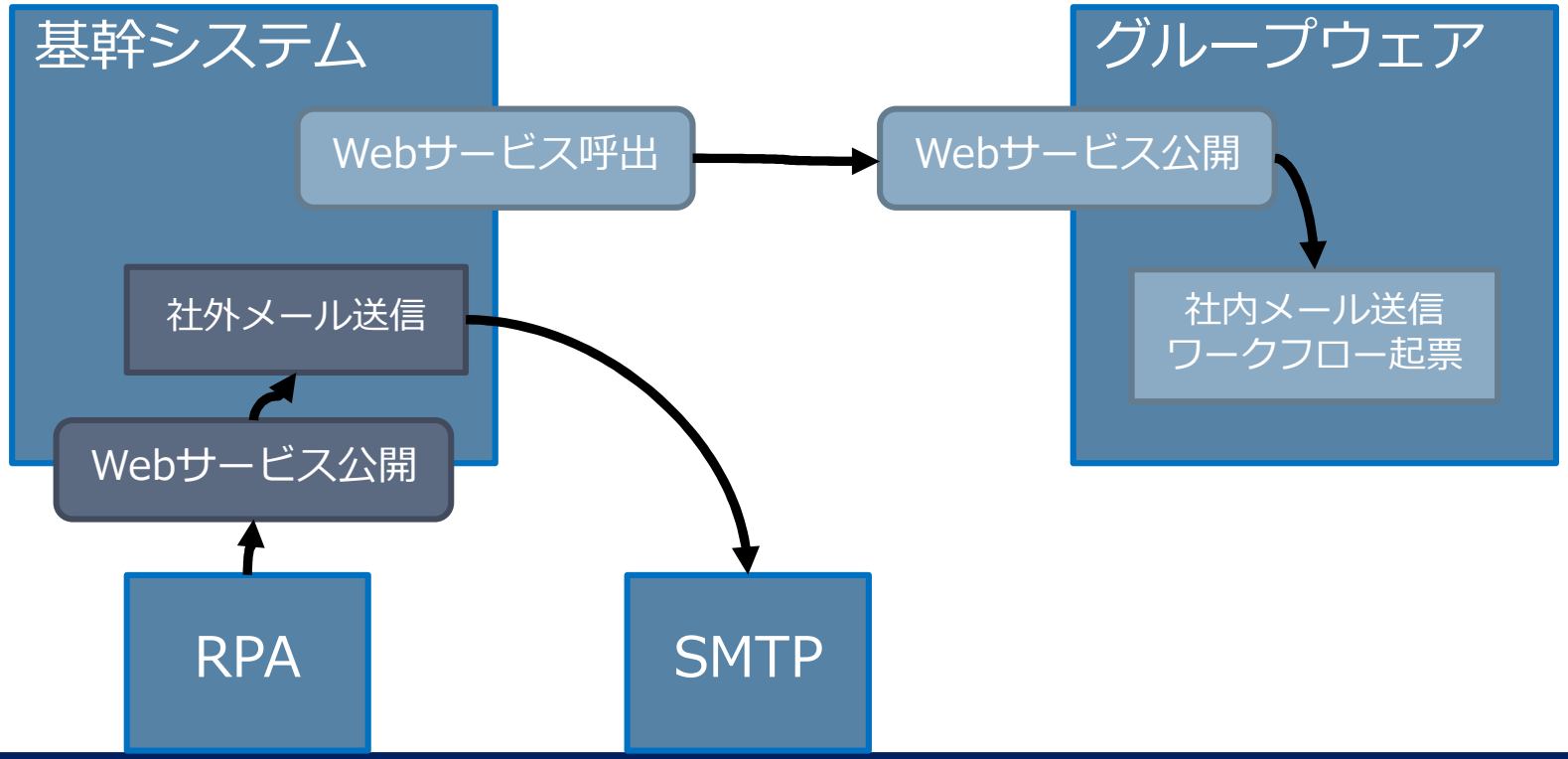
FTP通信、暗号/復号化などの基盤を整備し、実現可能となった



システムリフォーム（ユニチカモデル）であればこそ容易に実現できた事例③：Webサービス連携

- 社内メール、ワークフローシステムの起票、外部メールサービスの公開

Webサービス関連の基盤を整備し、実現可能となった

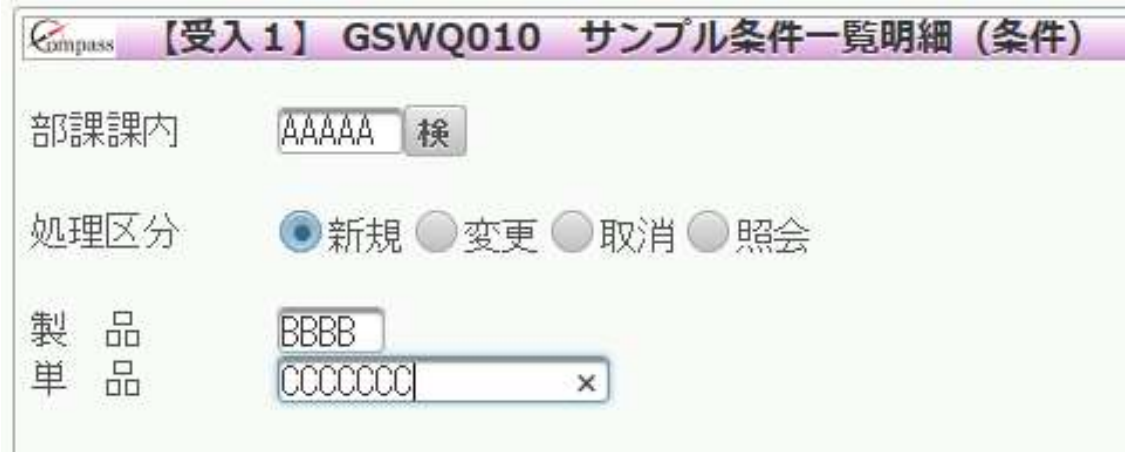


システムリフォーム（ユニチカモデル）であればこそ容易に実現できた事例④：大文字強制入力

- ユニチカのコード体系は半角アルファベット小文字「a,b,c,・・・」は使用せず、全て大文字「A,B,C,・・・」となっている。
- ホストの画面は大文字強制入力（**CAPSロックされていない状態でも大文字が入る**）となっていた。
- 本番開始後、利用部門よりCAPSキーをいちいち切り替えたりSHFTキーを押すのが面倒という問い合わせ。
- アプリケーション基盤を拡張し、

大文字強制入力を実現した。

全システムへ影響するため
かなりリスクのある変更



1. 今後の予定
（DX化　2025年の崖に向けて）

今後の予定（DX化　2025年の崖に向けて）

基盤の洗練・拡張

これまで、大規模な基幹システムを効率的に移行するためのベースラインアーキテクチャを構築してきましたが、今後、長期にわたって保守・拡張していける**エンタープライズな基幹システムに発展させるべく、アーキテクチャを洗練・拡張させていく**。

アプリケーション基盤構築の初期メンバーを抱えることで、本番開始後の拡張のリスクを軽減

準備期間
- アーキテクチャ原則策定
- アプリケーション基盤の基本機能の整備
- サンプルプログラムの整備
- 設計、開発ルールの整備

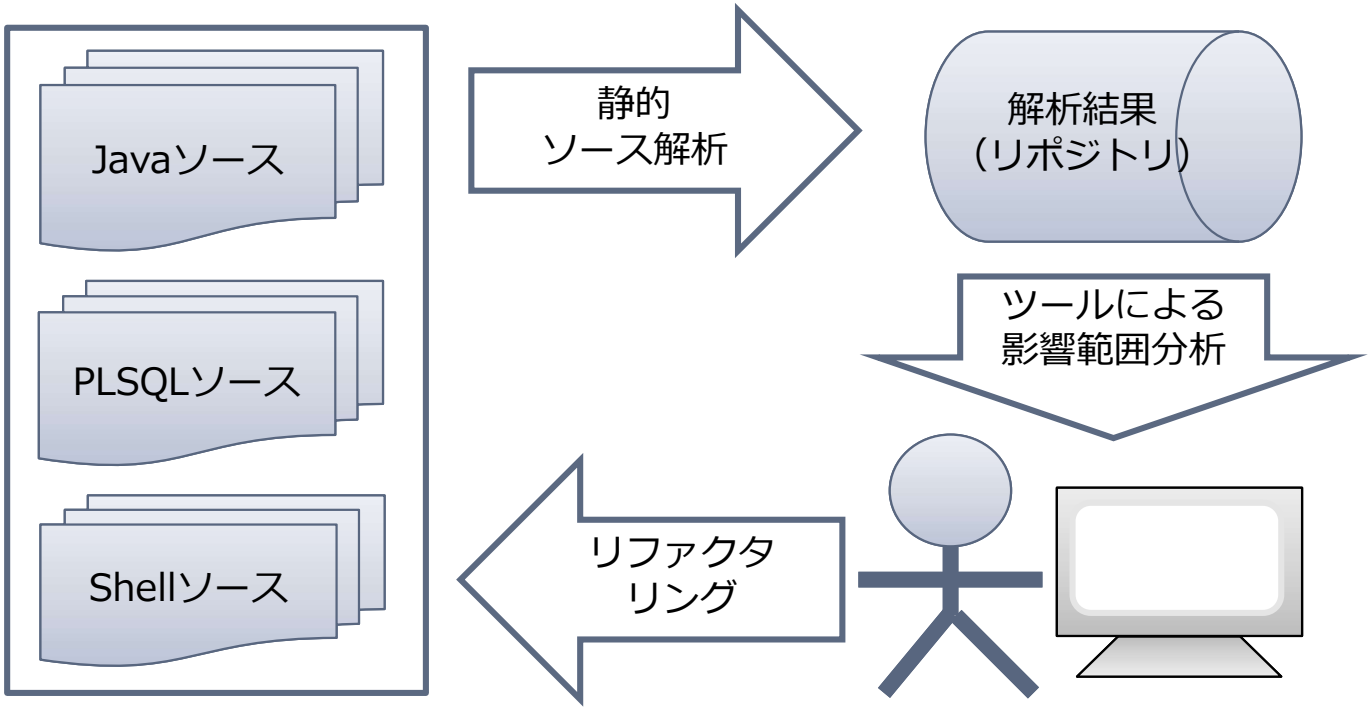
POC、本開発、ST
- アプリケーション基盤の不具合修正、不足機能追加

本番開始後
- **アプリケーション基盤の機能拡張**

今後の予定（DX化　2025年の崖に向けて）

リファクタリング

- **静的ソース解析の情報を利用し、利用されないロジックの削除や、部分的な置換えにより段階的に改善していく。**



・改善内容：

コードのシンプル化

変数スコープの限定

⇒ **マイクロサービス化**

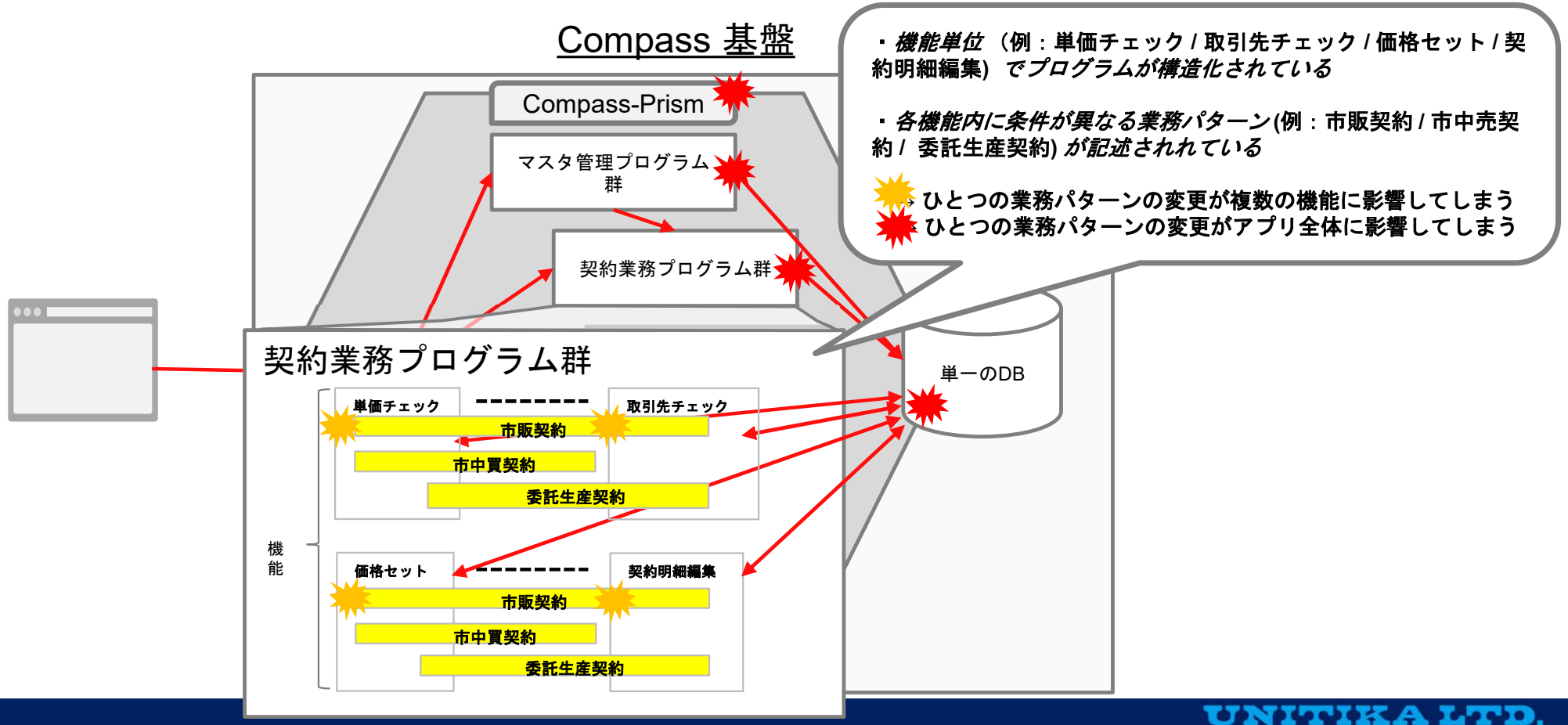
・リファクタリング目的：

パフォーマンス向上

保守性向上

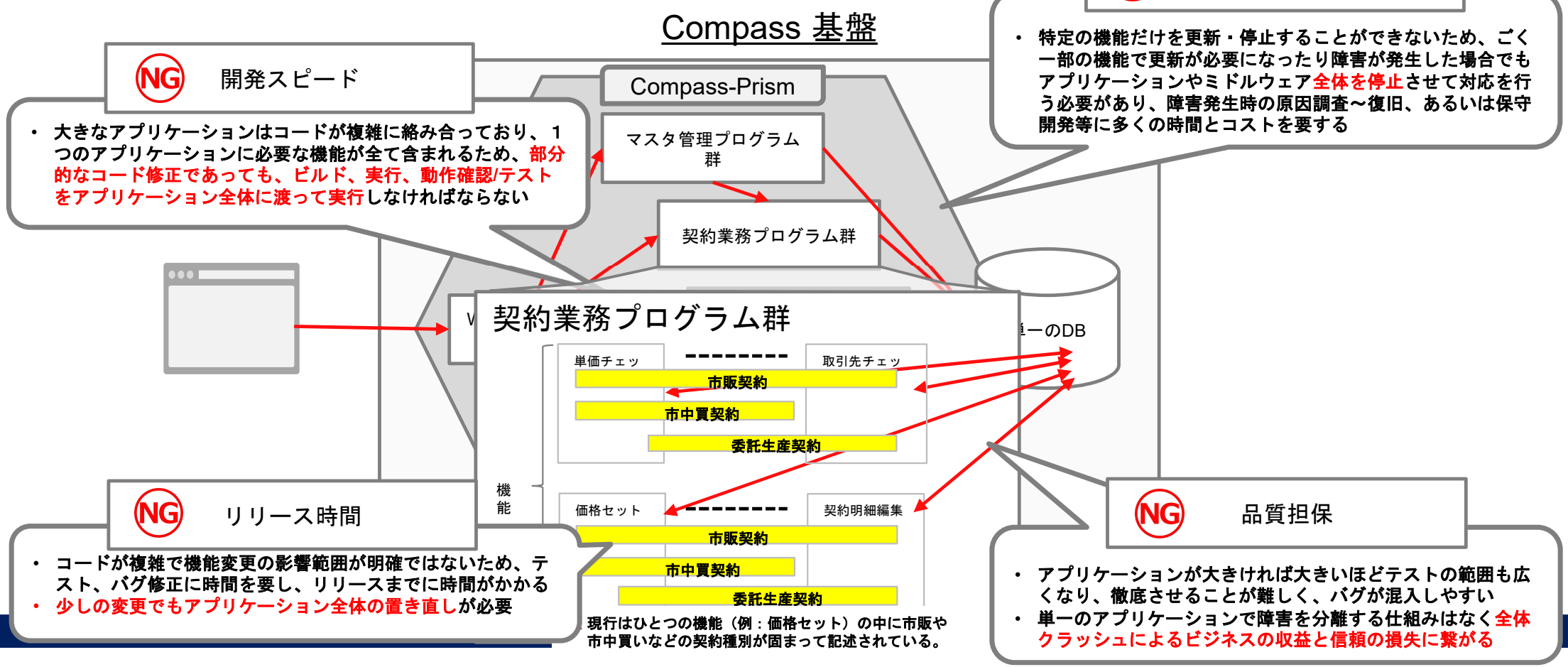
今後の予定（DX化　2025年の崖に向けて）

現状Compass 基盤上の実装イメージ



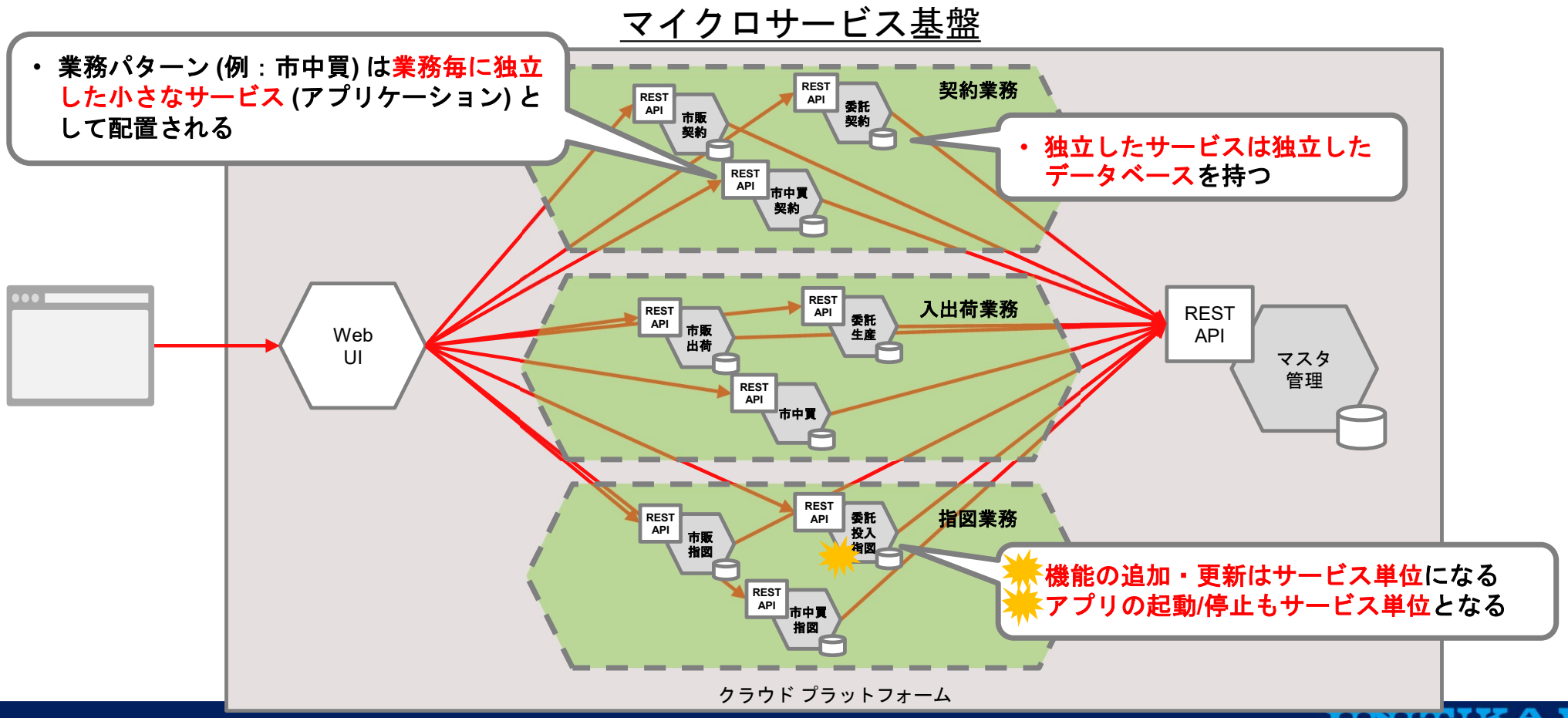
今後の予定（DX化　2025年の崖に向けて）

現状Compass が抱える開発・運用面での課題



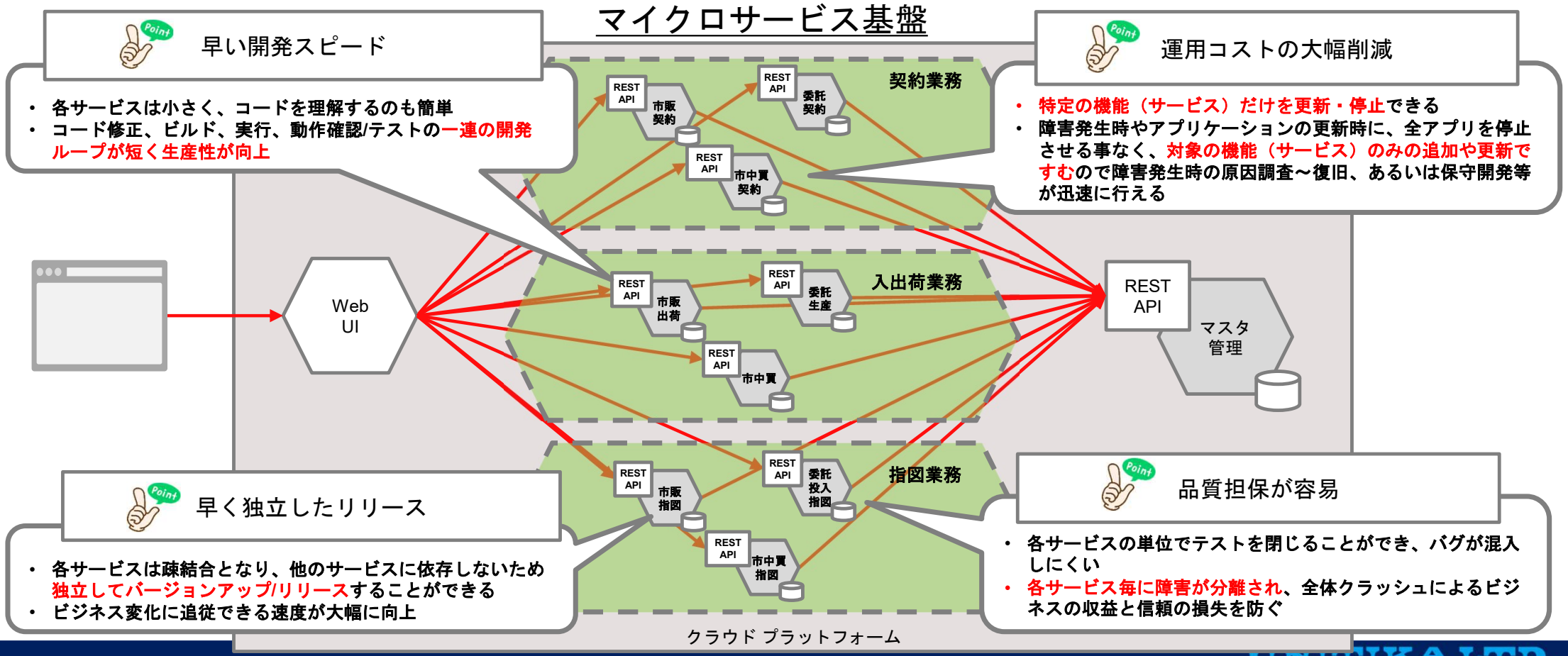
今後の予定（DX化　2025年の崖に向けて）

マイクロサービス基盤上の実装イメージ



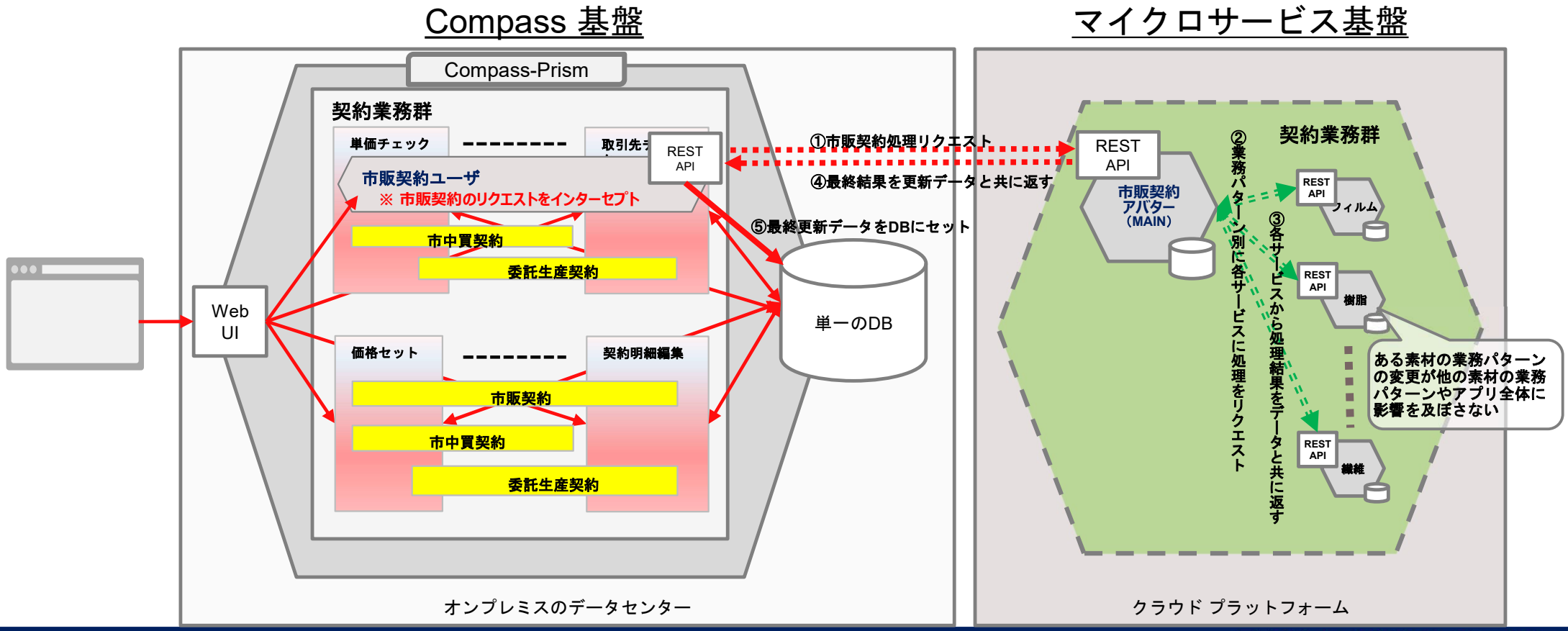
今後の予定（DX化　2025年の崖に向けて）

マイクロサービス基盤上の実装イメージ



今後の予定（DX化　2025年の崖に向けて）

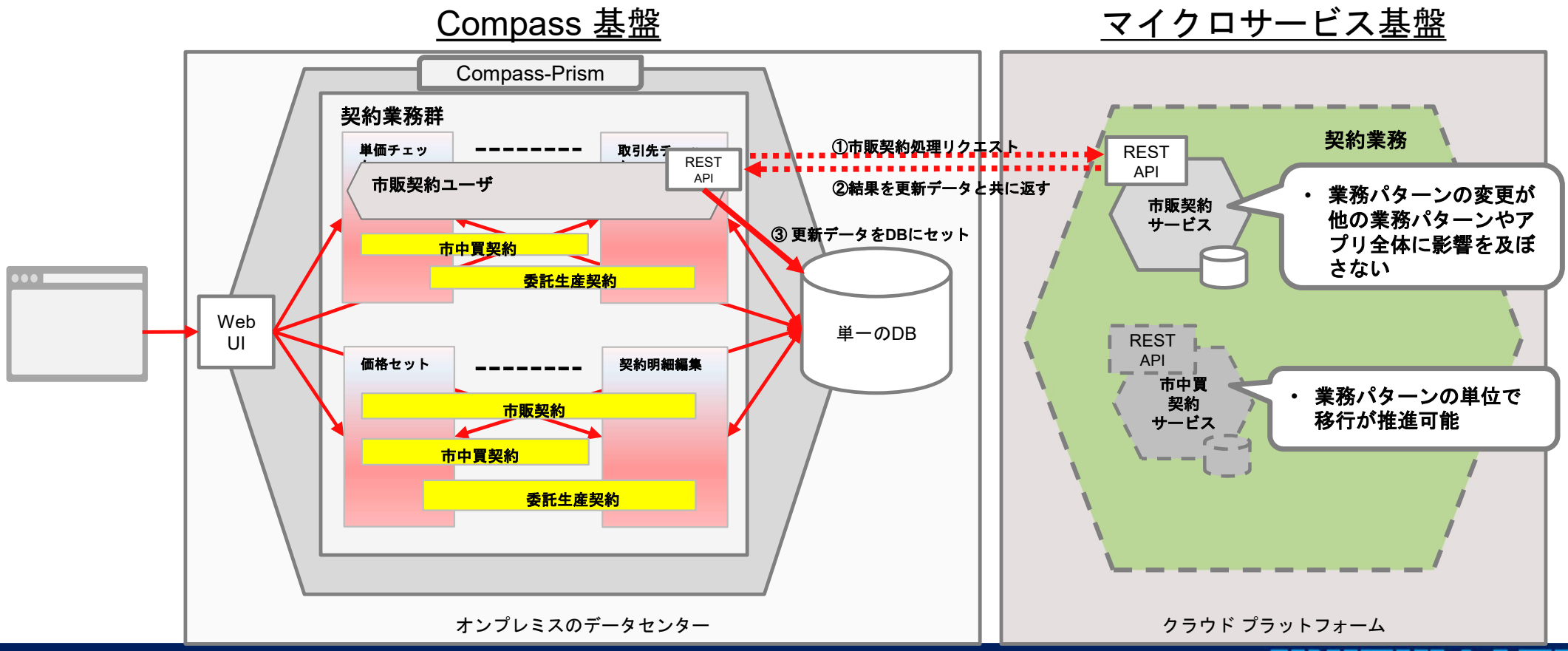
Compass 基盤とマイクロサービス基盤の併用



UNITIKA LTD.

今後の予定（DX化　2025年の崖に向けて）

Compass 基盤とマイクロサービス基盤の併用




技術×発想力
素材で未来をカタチに。

UNITIKA

We Realize It!

ご清聴ありがとうございました。