

The logo features the AWS logo on the left, consisting of the lowercase letters 'aws' in a white sans-serif font with a white curved arrow underneath. To the right of the AWS logo, the word 'SUMMIT' is written in a large, white, all-caps sans-serif font.

aws SUMMIT

TOKYO | APRIL 20-21, 2023

CUS-04

トランスフォーメーションの軌跡 ～ AWS とサーバーレス技術を活用した 物流のモダナイズ～

萩原 健介

ニッコン情報システム株式会社

ソリューションGr. グループマネジャー



自己紹介

萩原 健介 (はぎわら けんすけ)

経歴

- 組み込みエンジニアとして組み込み機器の開発
- 2017年 ニッコン情報システム株式会社 入社
- ニッコン情報システム株式会社 入社後からWEBシステムの開発に携わる

現在

- 当セッションでお話しするWMS(倉庫管理システム)構築プロジェクトにプロジェクトマネージャーの立場で参画中

会社紹介

会社名	ニッコン情報システム株式会社
所在地	東京都中央区入船3丁目1番13号 入船ニッコンビル4F
設立	2016年2月24日
資本金	5,000万円(2023年4月現在)
代表者	代表取締役社長 海野 克也
社員数	9名(2023年4月現在)
事業内容	総合物流企業であるニッコンホールディングスグループ各社の物流・流通加工のノウハウを最大限に活用した物流システムの構築

ニッコンホールディングスグループ紹介

ニッコンホールディングス株式会社(東証プライム市場)を中心とした総合物流企業グループ



国内拠点
45社

海外拠点
9カ国
29社

(国内事業所数 287事業所)

本セッションの概要

- ニッコンホールディングスグループの中核事業会社である「日本梱包運輸倉庫株式会社」とともに、WMS(倉庫管理システム)を構築し、2022年7月から運用を開始しました。
- WMS構築時に直面した課題と対応、今後の展望も交えてお伝えします。
- アマゾン ウェブ サービス(AWS)への移行を検討されている方、サーバーレスアーキテクチャの採用を検討されている方が、主な対象者となります。

アジェンダ

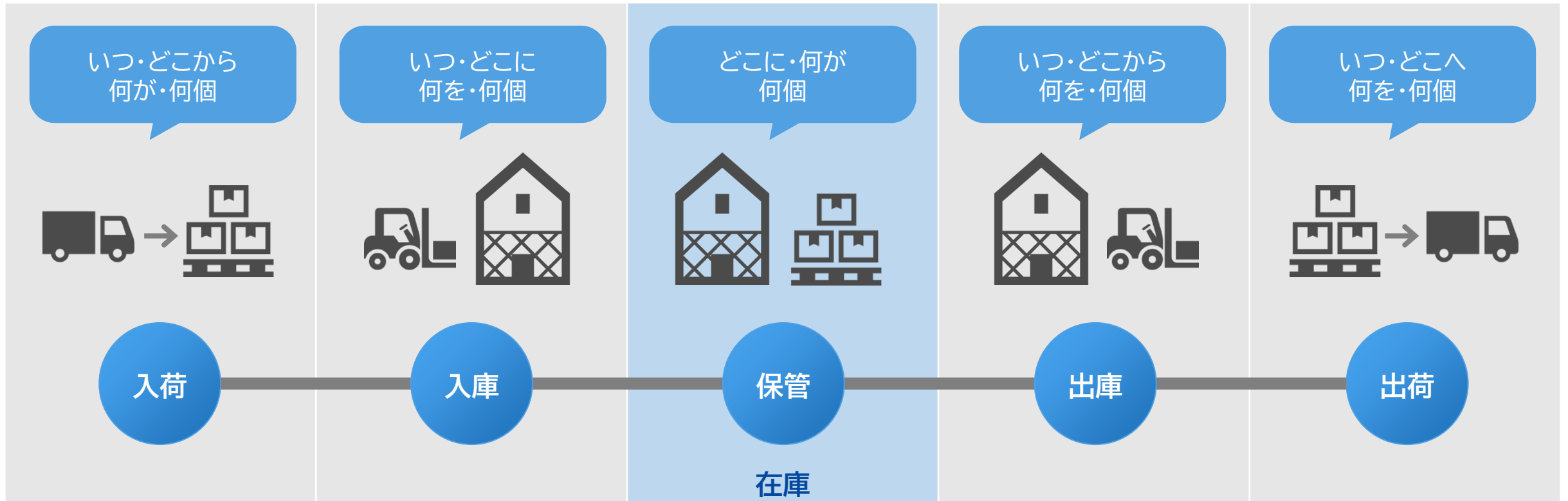
1. WMSとは
2. WMS開発に至る経緯
3. WMS全体像
4. AWS利用経験
5. WMS構築時の課題
6. 今後の展望
7. さいごに

アジェンダ

1. WMSとは
2. WMS開発に至る経緯
3. WMS全体像
4. AWS利用経験
5. WMS構築時の課題
6. 今後の展望
7. さいごに

WMSとは

Warehouse Management System の略称で、
倉庫管理における各工程で、商品の動きや在庫を管理するためのシステムです。



アジェンダ

1. WMSとは
- 2. WMS開発に至る経緯**
3. WMS全体像
4. AWS利用経験
5. WMS構築時の課題
6. 今後の展望
7. さいごに

WMS開発に至る経緯 – 課題

主に以下の課題を解消するためにWMSをフルスクラッチで構築することになった。

No.	課題	影響
1	複数のWMS(IBM AS/400)が稼働している(荷主ごとにカスタマイズ)	<ul style="list-style-type: none">改修までに時間がかかるメンテナンスコスト増加
2	文字中心の画面	<ul style="list-style-type: none">表示件数が少ない慣れるまでに時間がかかる(属人化)
3	紙中心の業務	<ul style="list-style-type: none">紙のコスト目視確認が必要(ミスの要因)
4	メールやFAXでの情報連携	<ul style="list-style-type: none">システムへ手入力

WMS開発に至る経緯 – 対策

主に以下の課題を解消するためにWMSをフルスクラッチで構築することになった。

No.	課題	対策
1	複数のWMS(IBM AS/400)が稼働している(荷主ごとにカスタマイズ)	<ul style="list-style-type: none">機能の汎用化工程ごとに必要な機能を選択
2	文字中心の画面	<ul style="list-style-type: none">操作の汎用化誰でも使える操作性
3	紙中心の業務	<ul style="list-style-type: none">アナログからデジタルへ転換ペーパーレス
4	メールやFAXでの情報連携	<ul style="list-style-type: none">アナログからデジタルへ転換自動連携による手入力の廃止

汎用化

効率化

WMS開発に至る経緯 – 効果

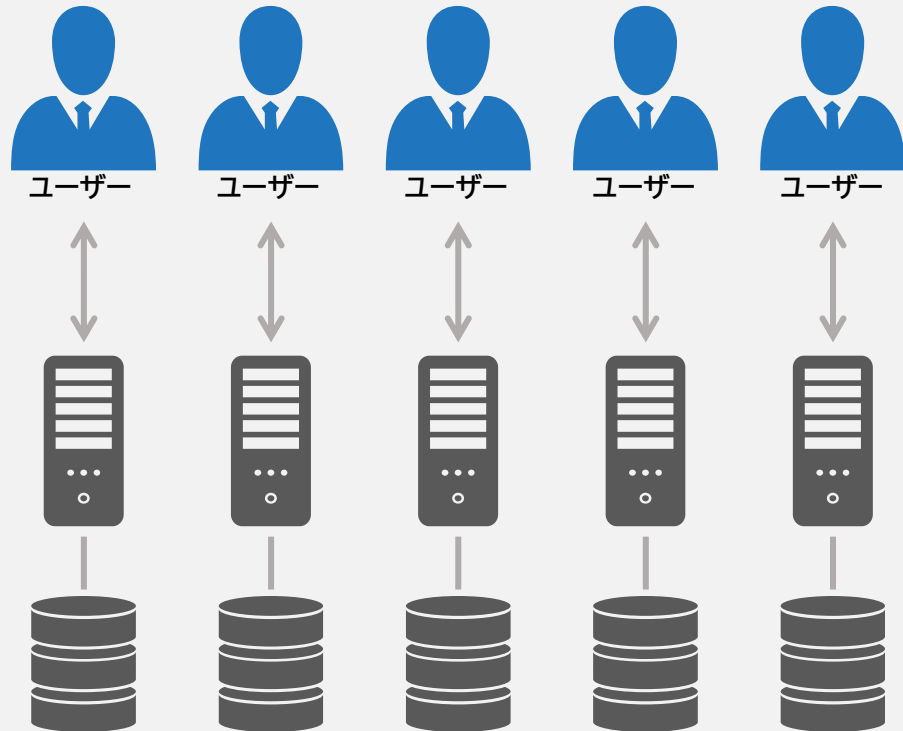
WMSを導入することで以下の効果を得られる。

No.	課題	効果
1	複数のWMS(IBM AS/400)が稼働している(荷主ごとにカスタマイズ)	<ul style="list-style-type: none">• 複数システムの使い分けが不要• メンテナンスコスト削減
2	文字中心の画面	<ul style="list-style-type: none">• 無駄な操作を削減• 属人性を排除
3	紙中心の業務	<ul style="list-style-type: none">• ペーパーレスによるコスト削減• 情報のリアルタイム性向上
4	メールやFAXでの情報連携	<ul style="list-style-type: none">• 自動連携による手入力の廃止

課題①

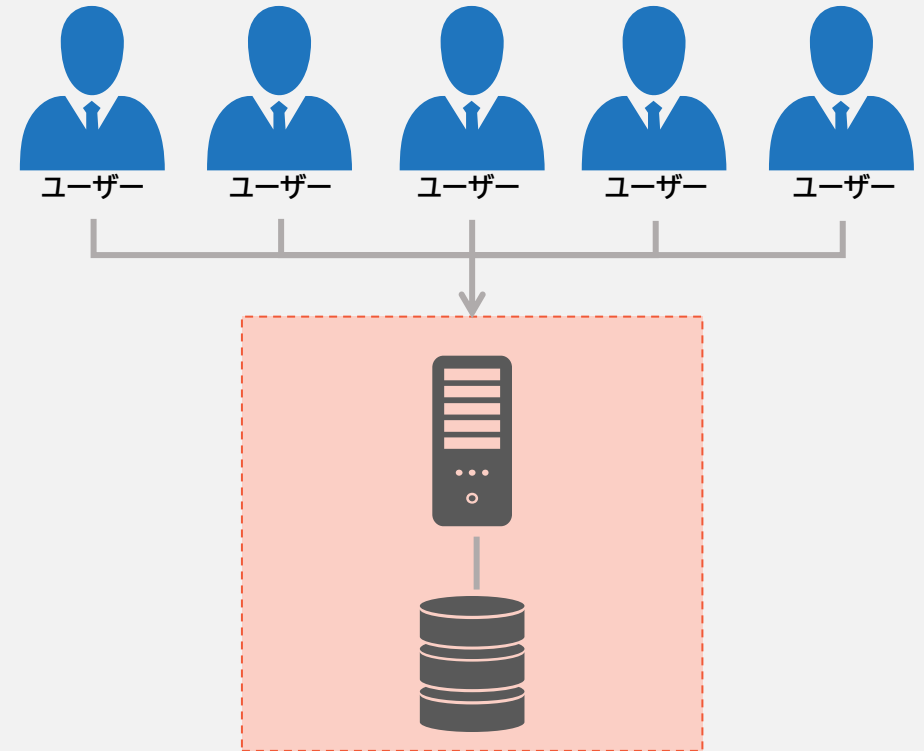
すべての荷主・業務ごとに必要な機能を実装・カスタマイズしていた。

これまで



荷主・業務ごとにシステムを用意

これから

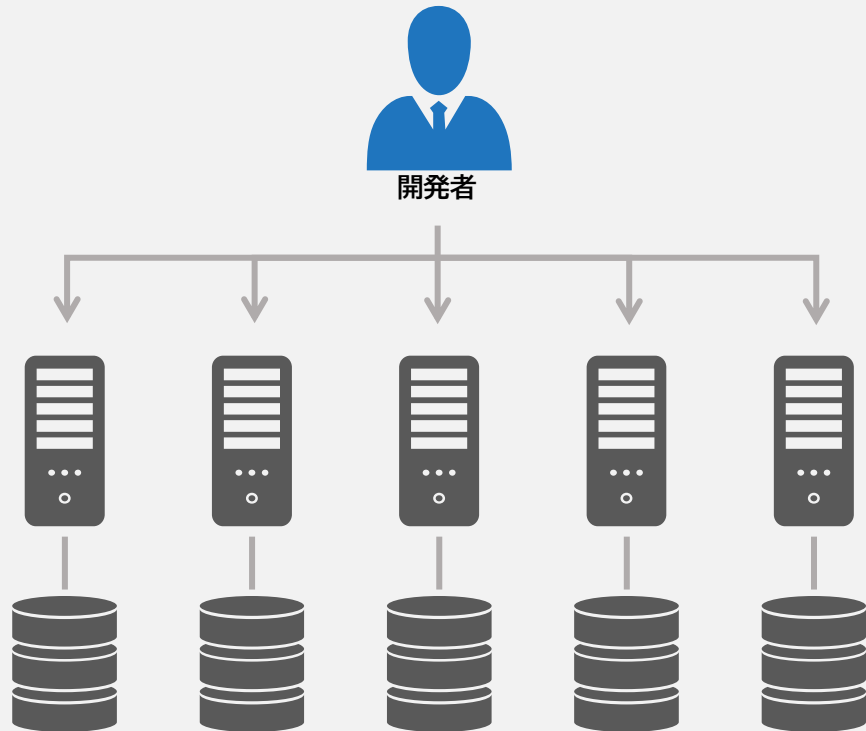


プラットフォームを一元化

課題①

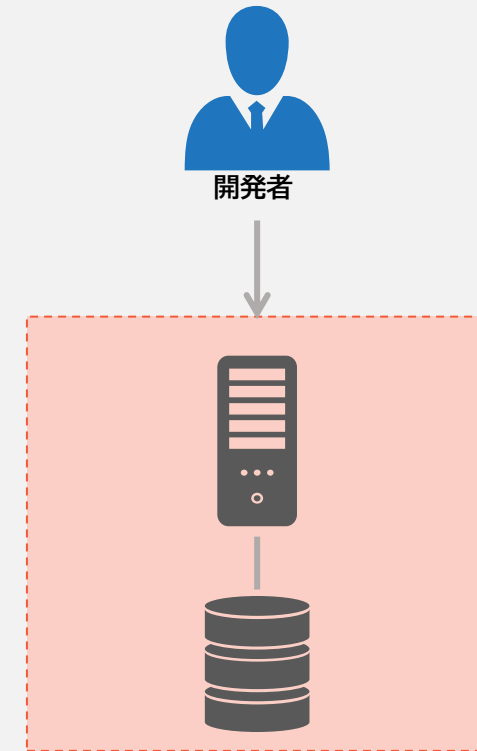
全システムのメンテナンスが必要だった。

これまで



荷主・業務ごとにシステムを用意

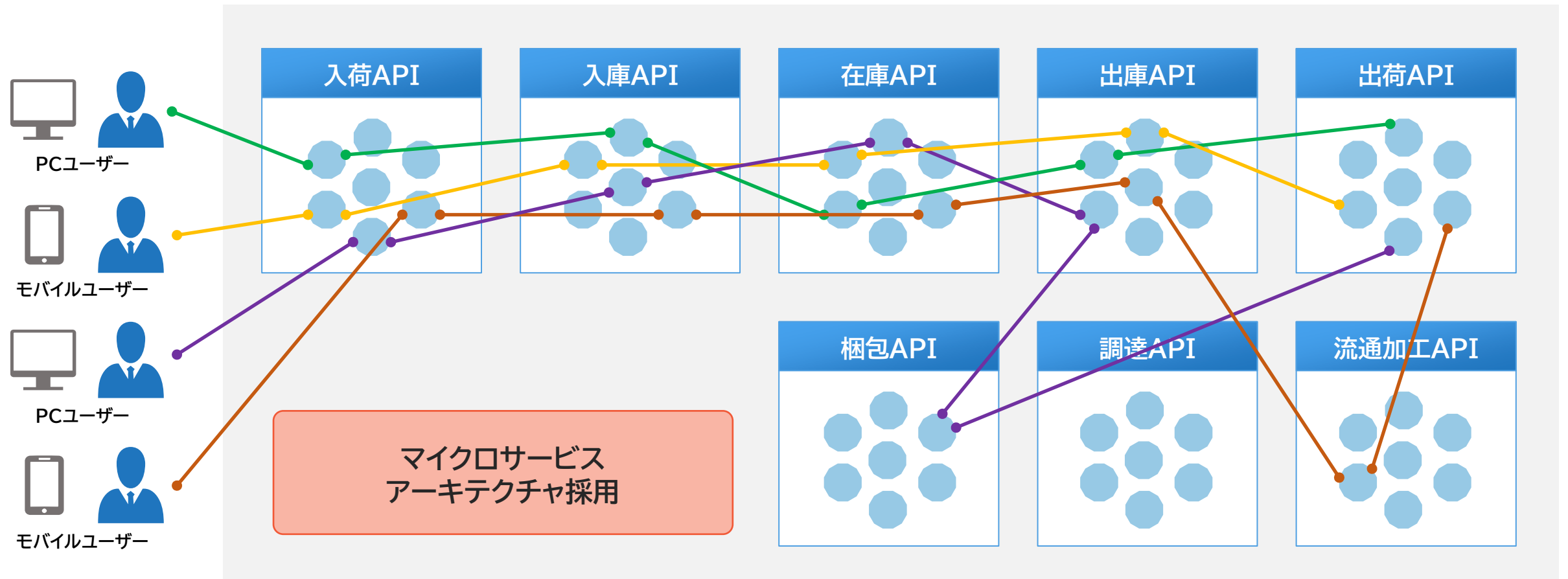
これから



プラットフォームを一元化

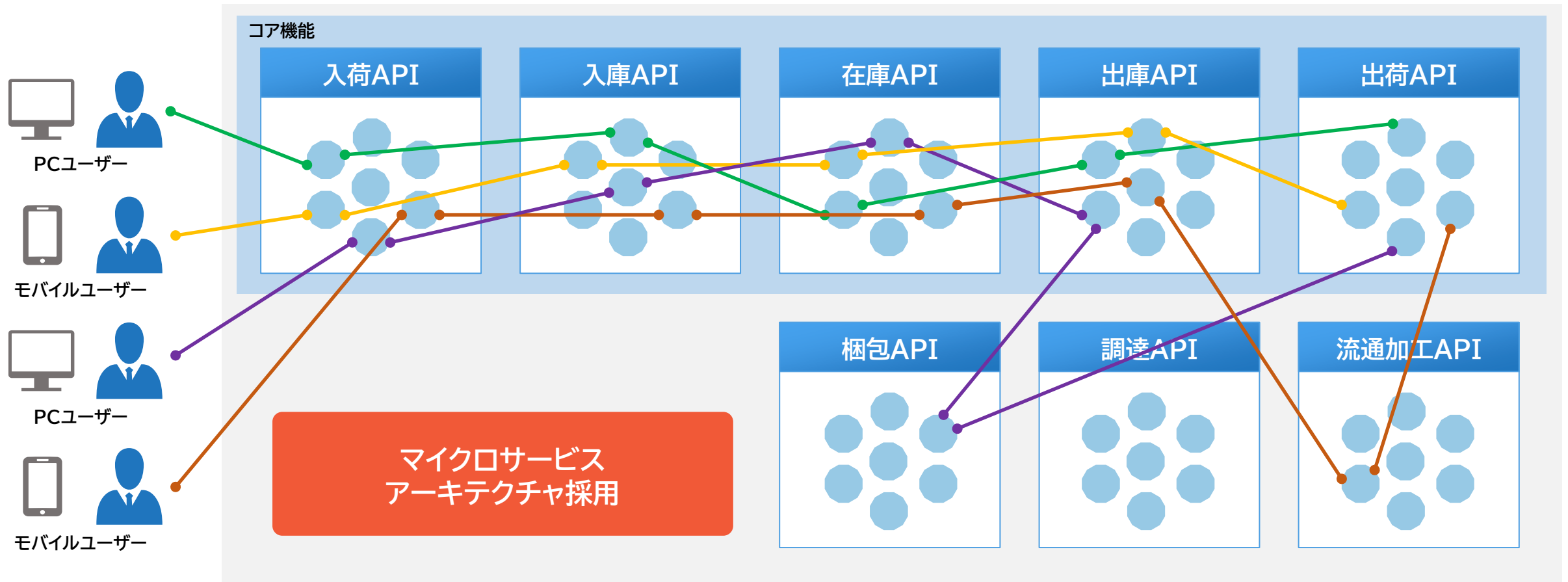
課題①の対策

汎用的なWMSを構築するには、各工程で業務に適したAPIを選択できる必要がある。



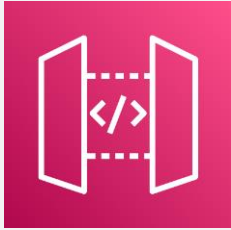
課題①の対策

必要な機能を後から追加する。必要なときに必要な機能だけをリリースする。



課題①の対策

マイクロサービスを採用すると開発チームの数が増える & 開発チームごとの環境が必要
システム(アプリ)の開発に注力したい → インフラ構築の工数を削減が必要
サーバーレス技術により、インフラ構築の工数を削減する。



Amazon API Gateway



Amazon Elastic Container Registry (Amazon ECR)



AWS Fargate



AWS Lambda



Amazon Elastic Container Service (Amazon ECS)

マイクロサービス・サーバーレスと
親和性が高いサービス

課題②

誰でも使える操作性

これまで

CUI

(キャラクターユーザインタフェース)



```
H: ¥ > cd Documents  
H: ¥ Documents > dir
```

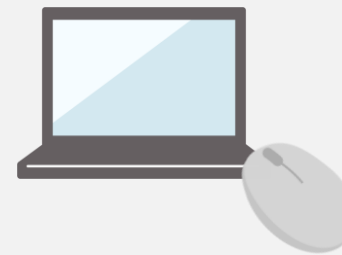
キーボードでのコマンド入力によって
操作を行う

習熟が必要

これから

GUI

(グラフィカルユーザインタフェース)



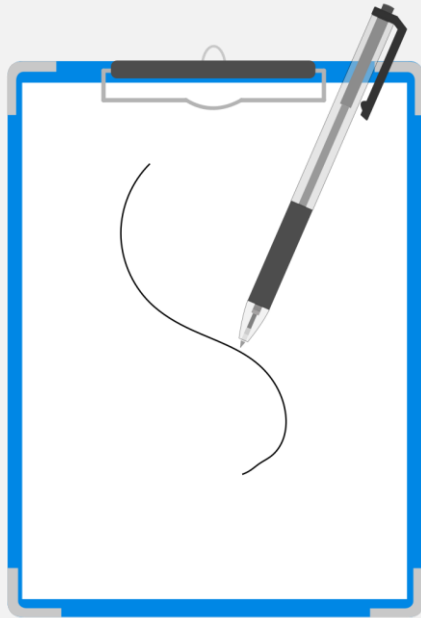
マウスやタッチパネルを使い
アイコン・ボタンなど視覚的な操作を行う

誰でもすぐに使える

課題③

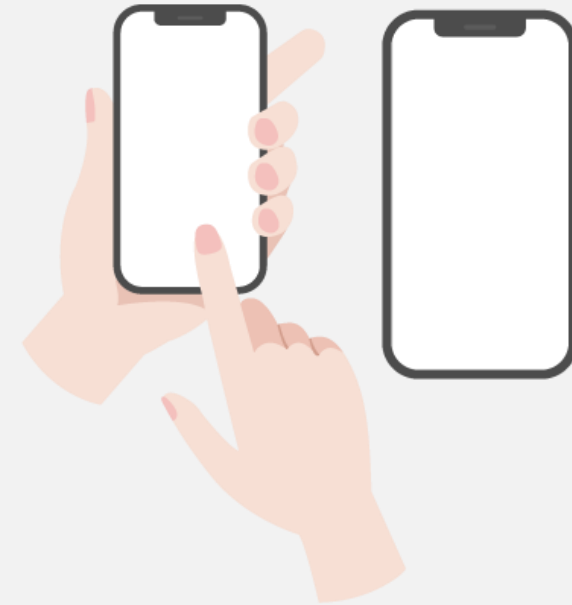
構想:ペーパーレス化・情報のリアルタイム制

これまで



紙に手書き

これから

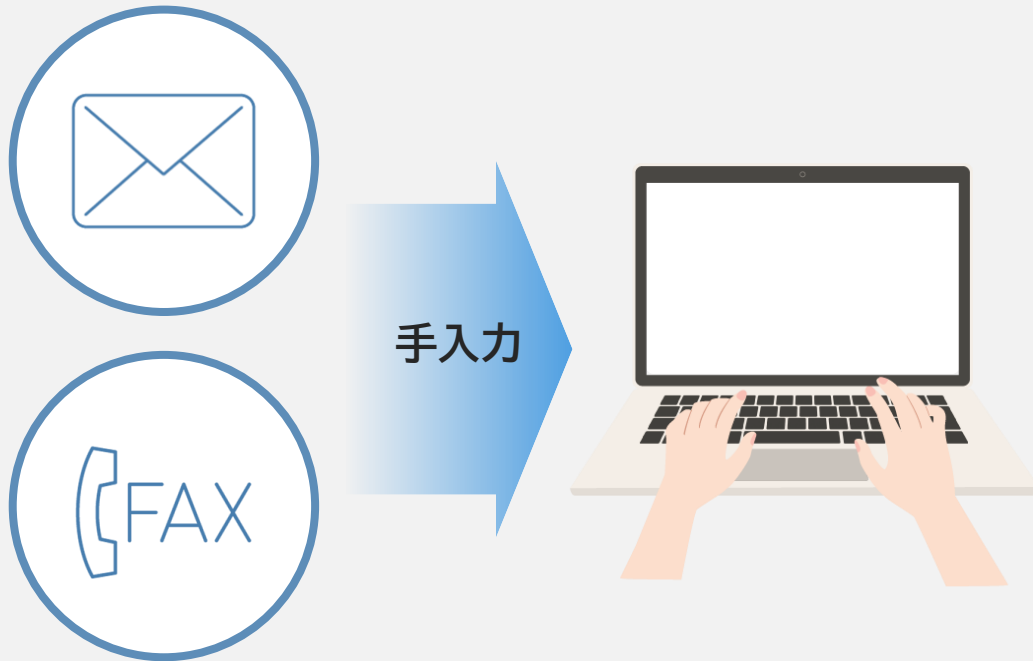


ハンディ端末

課題④

構想:情報の連携

これまで



メールやFAXで情報を受信して手入力

これから

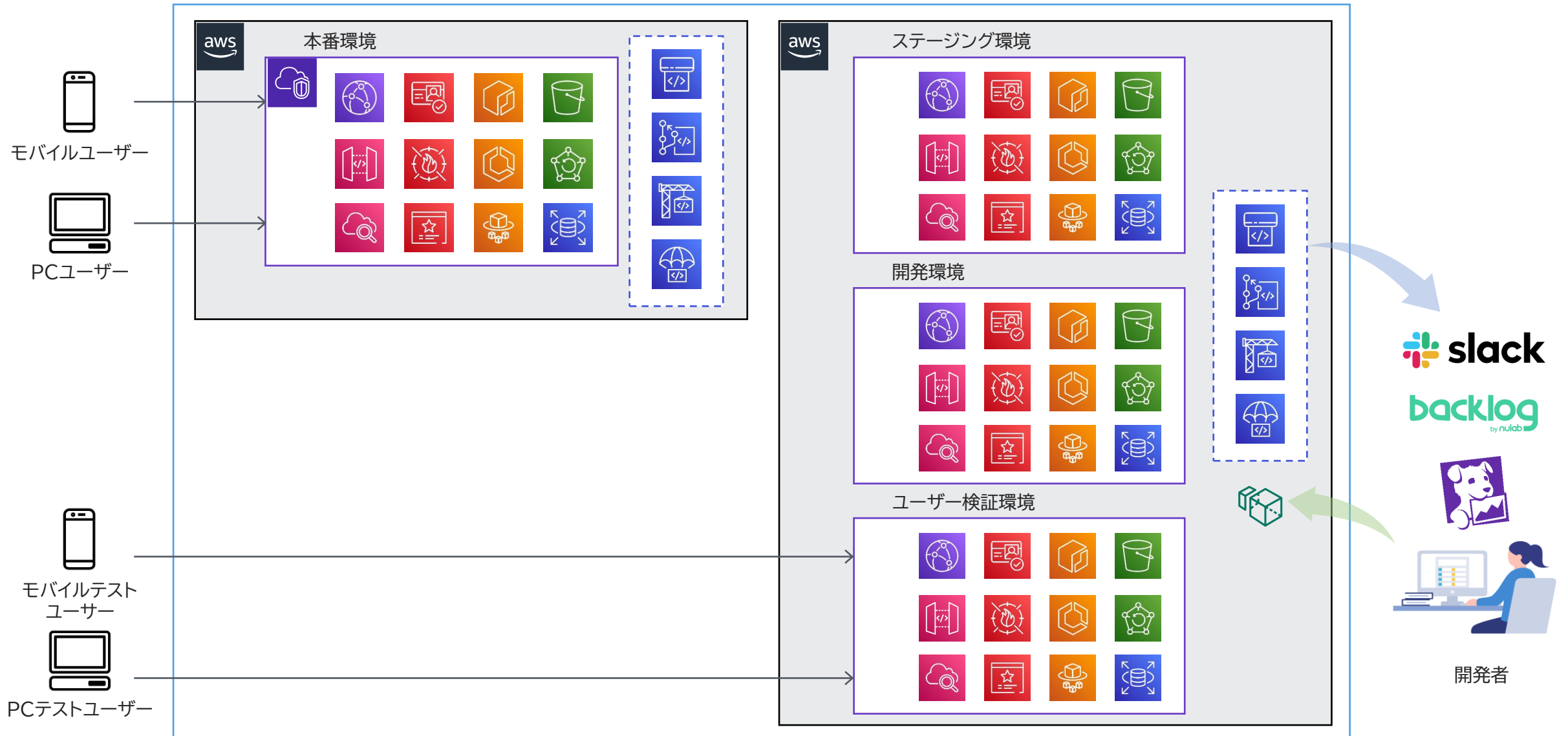


連携により自動入力

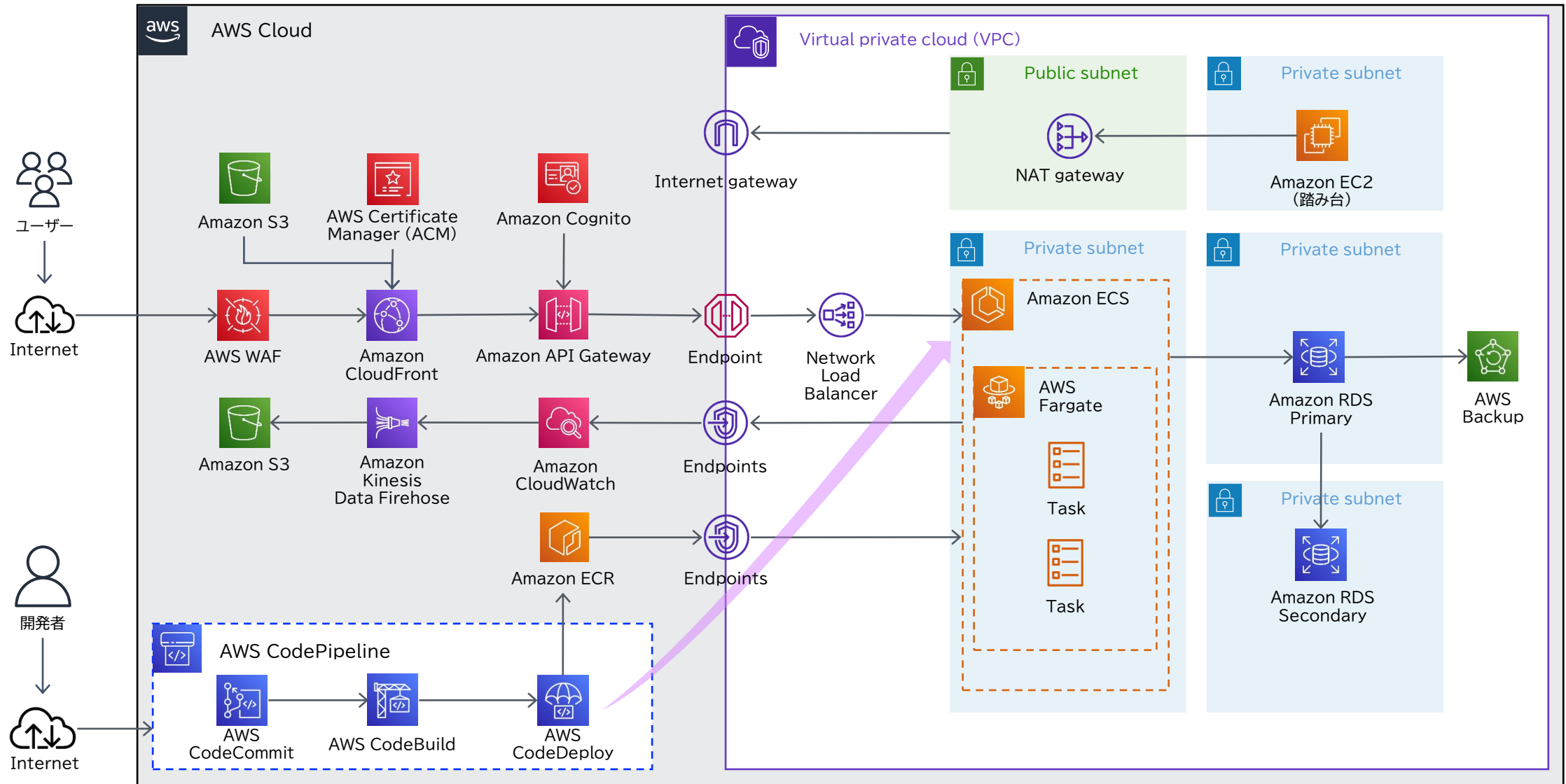
アジェンダ

1. WMSとは
2. WMS開発に至る経緯
- 3. WMS全体像**
4. AWS利用経験
5. WMS構築時の課題
6. 今後の展望
7. さいごに

WMS全体像一構成図(全体)



WMS全体像一構成図(本番環境)



WMS全体像ーAWSサービス・リソース

WMSで利用した主要AWSサービス・リソース

- AWS CloudTrail
- AWS Config
- Amazon CloudWatch

- Amazon Kinesis Data Firehose
- Amazon S3

- AWS WAF
- AWS Certificate Manager
- Amazon Cognito
- Amazon CloudFront
- Amazon API Gateway

- Virtual private cloud (VPC)
- Internet gateway
- Peering connection
- Network Load Balancer
- Endpoints
- NAT Gateway

- Amazon ECS
- Amazon ECR
- AWS Fargate
- Amazon RDS
- Amazon EC2
- AWS Lambda

- AWS CodePipeline
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeDeploy

- Amazon SNS
- Amazon WorkSpaces

WMS全体像－開発環境・ツール

稼働環境	AWS
アーキテクチャ	サーバーレス・マイクロサービス
構成要素	Webアプリケーション Androidネイティブアプリケーション
使用言語・フレームワーク	Java / Spring Boot / Vue.js / JavaScript / Kotlin
開発ツール	Visual Studio Code/ Eclipse / Android Studio
コラボレーションツール	Slack / Backlog
運用監視ツール	Datadog



WMS全体像－主要機能

2022年7月運用開始時のWMS主要機能

【入荷・入庫】

- 入荷予定登録
- 入荷予定取込
- 入荷予定修正
- 入荷予定一覧
- 入荷実績登録
- 入荷実績一覧
- 入荷実績報告
- 入庫指示一覧
- 入庫実績登録

【在庫・保管】

- 在庫照会
- 在庫推移
- 在庫報告書
- 在庫情報変更指示登録
- 在庫情報変更指示修正
- 在庫情報変更指示一覧
- 在庫情報変更実績登録
- はい替え指示登録
- はい替え指示一覧
- はい替え実績登録

【出庫・出荷】

- 出荷予定登録
- 出荷予定取込
- 出荷予定修正
- 出荷予定一覧
- 出庫実績登録
- 出庫実績一覧
- 出荷実績登録
- 未出荷一覧
- 出荷実績報告

【その他】

- 棚卸計画登録
- 棚卸計画一覧
- 棚卸実績登録
- 入出庫履歴
- 実績確認
- 在庫トレース
- 請求
- 返品

WMS全体像－AWS採用のメリット

AWSへ移行することで得られるビジネス面でのメリット

1

インフラコストの削減

3

メンテナンス(運用・保守)コストの削減

2

サーバー機器等の調達工数削減

4

迅速な開発・迅速なリリース

アジェンダ

1. WMSとは
2. WMS開発に至る経緯
3. WMS全体像
- 4. AWS利用経験**
5. WMS構築時の課題
6. 今後の展望
7. さいごに

AWS利用経験

WMS構築前にAWSを利用した経験は以下の通りです。

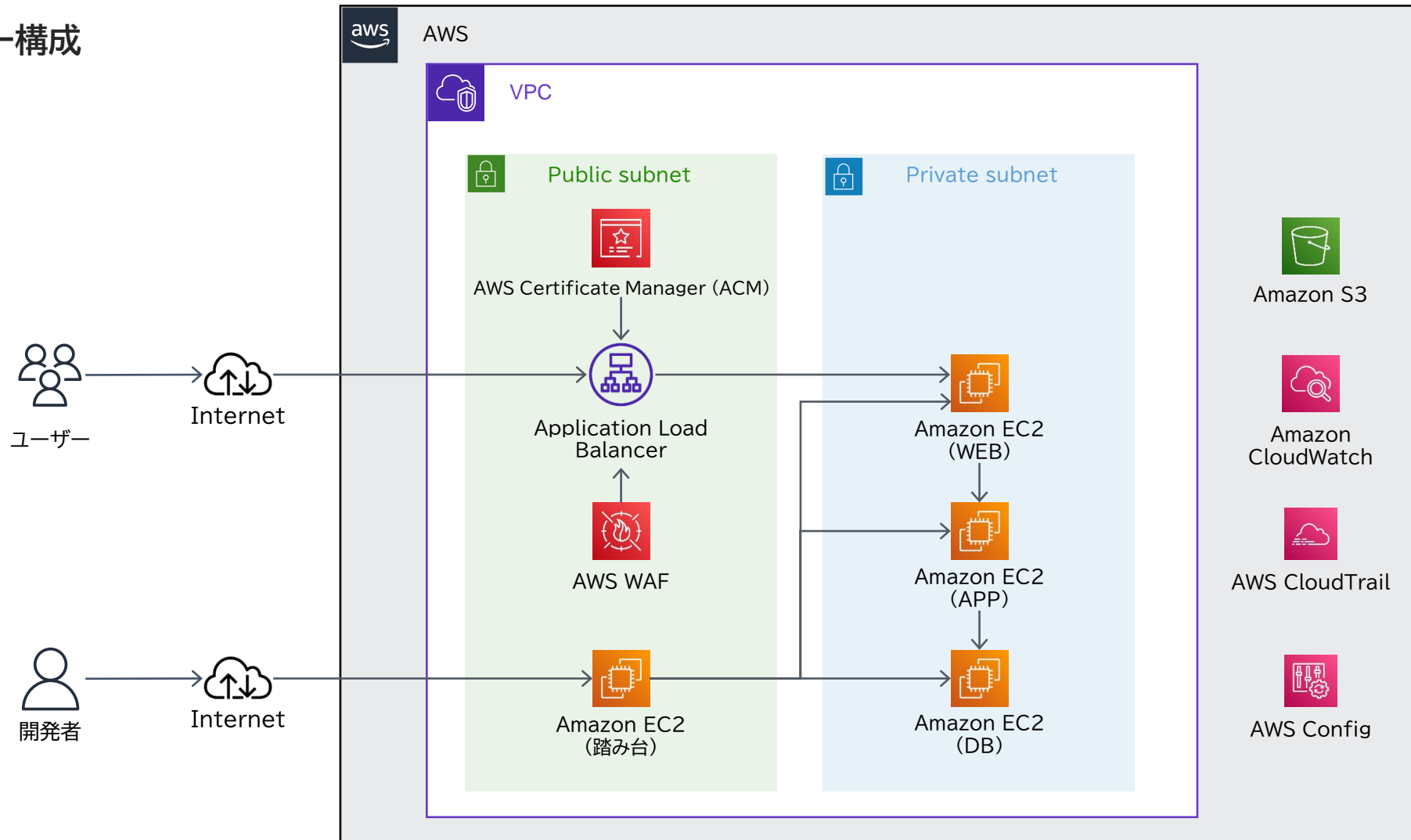
- Windows Server上にバージョン管理システム SVN を導入
- Windows Server上にプロジェクト管理ツール Redmine を導入
- WEBサーバー/APPサーバー/DBサーバー構成でシステム構築
(3層サーバー構成で1ヶ月あたりの利用料は5万円に達することはない。)
- 開発パートナーとともにサーバーレスアーキテクチャを用いてシステム構築にチャレンジ



WMS構築に着手

AWS利用経験

3層サーバー構成










アジェンダ

1. WMSとは
2. WMS開発に至る経緯
3. WMS全体像
4. AWS利用経験
- 5. WMS構築時の課題**
6. 今後の展望
7. さいごに

WMS構築時の課題ープロジェクト概要

開発期間	' 22年1月~6月末 ' 22年7月1日運用開始
開発規模	160人月程度

22年1月	22年2月	22年3月	22年4月	22年5月	22年6月	22年7月
開発前期			開発後期			リリース後
20人	20人	30人	30人	25人	25人	20人
						

短期での基盤整備

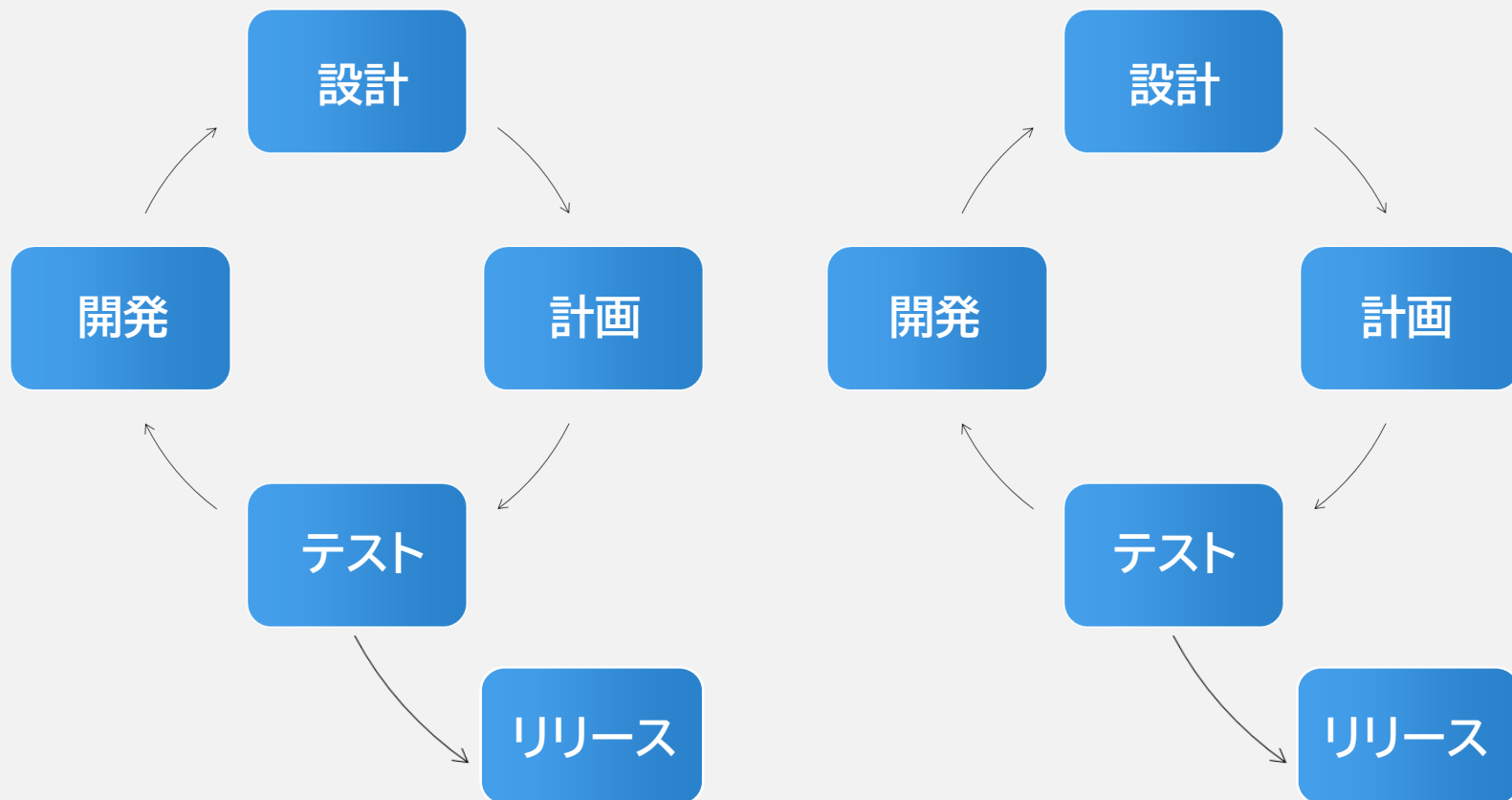


サーバーレス環境

WMS構築時の課題－開発手法

アジャイル手法を用いたMVP開発

※MVP(Minimum Viable Product)



MVP開発

効果の刈り取り

アジャイル手法

手戻り防止

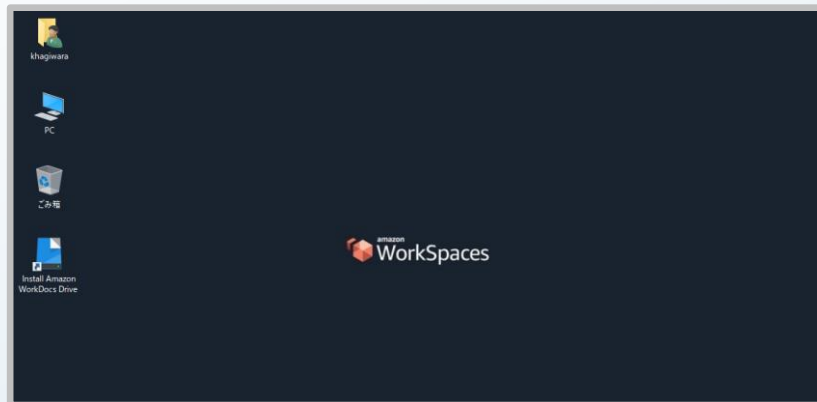
WMS構築時の課題①ー前期

課題

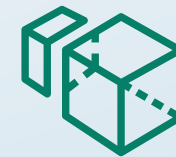
開発環境(PC)の準備

Amazon WorkSpaces

- 接続元のPCに制限がない
- 開発ツール導入済みの環境を事前準備できた。
- 新型コロナウイルスの影響によるテレワーク対応が容易だった
- WorkSpaces上にファイルサーバーをたて、ファイルを管理



ローカルPC上のAmazon WorkSpacesクライアント

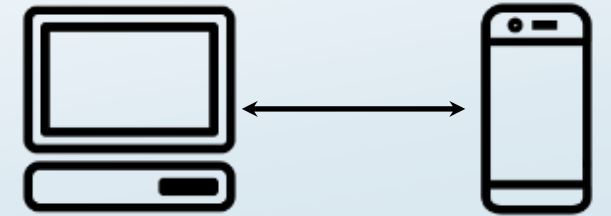


Amazon WorkSpaces

WMS構築時の課題②一前期

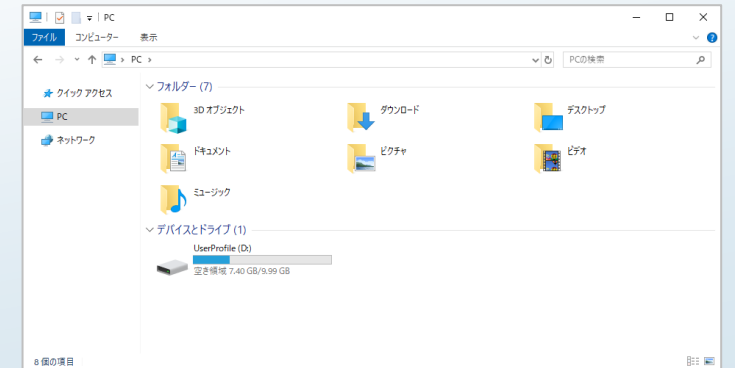
Amazon WorkSpaces(課題)

- WMSではモバイル端末にAndroidを採用している。
- Androidアプリ開発時にはモバイル端末とPCを有線接続する必要があった。
- Androidアプリ開発用にPCを用意。



Amazon WorkSpaces(課題)

- 仮想マシンのスペックをアップする際にCドライブがクリアされる。
(Dドライブのユーザーデータはそのまま残る。)



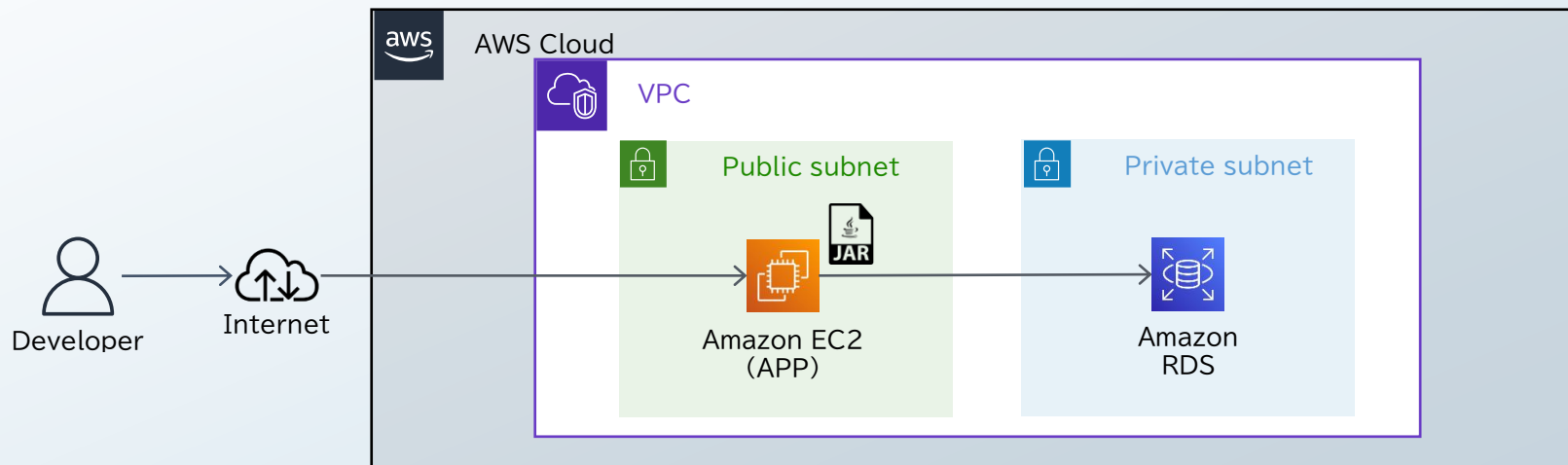
WMS構築時の課題③一前期

開発に着手したばかりなので、システムの稼働環境が準備できていない。

そのため、開発者のローカルPC上での確認が中心となるが、フロントエンド、モバイルアプリ、バックエンドを結合したテスト環境が必要だった。

スケジュールが短いなかで、サーバー調達の遅れは致命的だったが、AWSを使ったことで必要なときにサーバーをたてることができた。

とはいえ、サーバーレス環境の構築にはある程度の時間が必要だったので、当初はAmazon EC2上でJARファイルを起動して、結合テストを実施していた。



WMS構築時の課題④一後期

開発後期に入るとAPIが量産され、みるみるAPIの数が増加していった。

APIが増えるたびにAmazon API Gatewayへの登録する必要があった。

APIひとつひとつマネジメントコンソールへ手入力する必要があるので、100%完璧に登録することは困難だった。

(200程度のAPIがある。1つ1つマネジメントコンソールから手入力することで、漏れや入力違いが発生した。)

これを解決する手段として Infrastructure as Code(IaC)がある。

現在導入推進中。

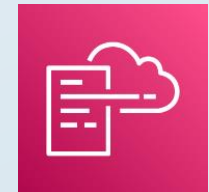
Terraformを使用する方向で推進しているが、

AWS CloudFormationやAWS Cloud Development Kit (AWS CDK)

を取り入れることも検討している。



AWS Cloud Development Kit
(AWS CDK)



AWS CloudFormation

WMS構築時の課題⑤—後期

REST API の実行に関する Amazon API Gatewayクォータ

- ペイロードサイズ : 10MB
- 最大統合タイムアウト : 30秒

制限に達すると、Amazon API Gatewayにより接続が切断される。

- Amazon API Gatewayを介さない通信手順を導入する。
- ペイロードを分割して送受信する。

WMS構築時の課題⑥—後期

Amazon Cognitoによるアカウントロック

- ▶ Amazon Cognito コンソール上からはロックアウトしたユーザーを確認できない。
- ▶ AWS CloudTrail からであれば確認できるが、ロックされたユーザーを特定できない。
- ▶ ログイン画面の戻り値(エラーメッセージ)で判定する。

エラーメッセージ

- ユーザーID・パスワード違い :Incorrect username or password.
- アカウントロック :Password attempts exceeded.

WMS構築時の課題⑦ーリリース後

オートスケーリング

- ▶ スケールアウトやスケールインによる負荷に応じたタスク数の増減が容易
- ▶ スケーリングポリシーの選定と設定値の決定が重要

負荷が高いときにスケールアウトしなかったり、負荷が低いときにスケールインしない。

スケーリングポリシー

- ターゲット追跡スケーリング
- ステップスケーリング
- スケジュールに基づくスケーリング

WMS構築時の課題⑧ーリリース後

ヘルスチェック

- ▶ AWS Fargateで起動していても応答がないタスクが存在することがある。
- ▶ 各コンテナのタスク定義にて、ヘルスチェックのコマンドを定義することができる。
- ▶ パラメータも含めた実際のリクエストを使用して、ヘルスチェックを行うことで、動作不良のタスクをAmazon ECSが停止して、新しいタスクを起動してくれる。

アジェンダ

1. WMSとは
2. WMS開発に至る経緯
3. WMS全体像
4. AWS利用経験
5. WMS構築時の課題
- 6. 今後の展望**
7. さいごに

今後の展望

今後は、WMSを中心とした物流プラットフォームの構築に注力してゆく。

Cirrus / Platform

LMS(物流管理)

WMS(倉庫管理)

/ PL(調達物流)

/ PK(梱包)

/ DP(流通加工)

/ IM.EX(輸出入)

/ DC(機器制御)

TMS(輸送管理)

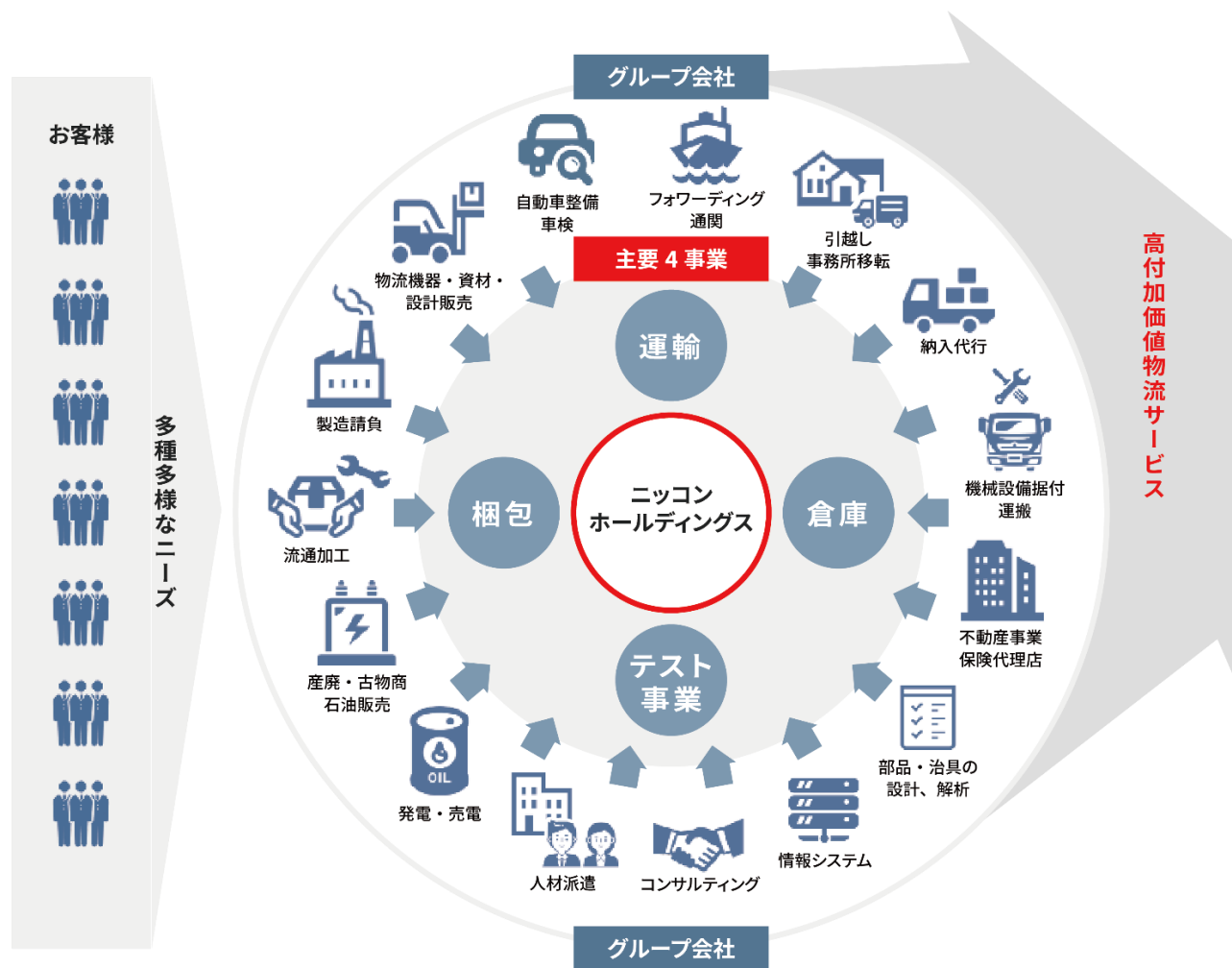
MDM(マスタ管理)

アジェンダ

1. WMSとは
2. WMS開発に至る経緯
3. WMS全体像
4. AWS利用経験
5. WMS構築時の課題
6. 今後の展望
- 7. さいごに**

さいごに

ニッコンホールディングスグループの各社は、
保管から梱包・加工、輸送を含めた
総合物流サービスを展開しています。
システム利用を含めたトータルサービスを
ご提供いたしますので、ご照会ください。



Thank you!

