UZABASE

2021/12/18

事業と共に育てる機械学習システムの これまでとこれから

小副川 健 (Takeshi OSOEKAWA)

We guide business people to insights that change the world

自己紹介

小副川 健 (Takeshi OSOEKAWA)

- 2018 年 UZABASE 入社
  - B2B SaaS事業 Fellow /
    SPEEDA CDS（Chief Data Scientist）
- Code for Japan Fellow
- 前職は SIer のデータサイエンティスト
- 大学時代は計算機代数学が専門

UZABASE

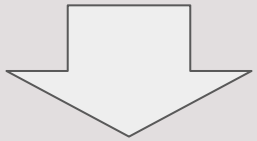
Code for Japan
ともに考え、ともにつくる。

目次

- 前口上: UZABASE のデータサイエンティストになるまで
- これまで
  - 構成管理
  - Python コンポーネントづくり
  - 機械学習周辺の話
- これから
  - 推論結果の小さなリリース
  - データメンテナンスのためのデータサイエンス

前口上: UZABASE のデータサイエンティストになるまで

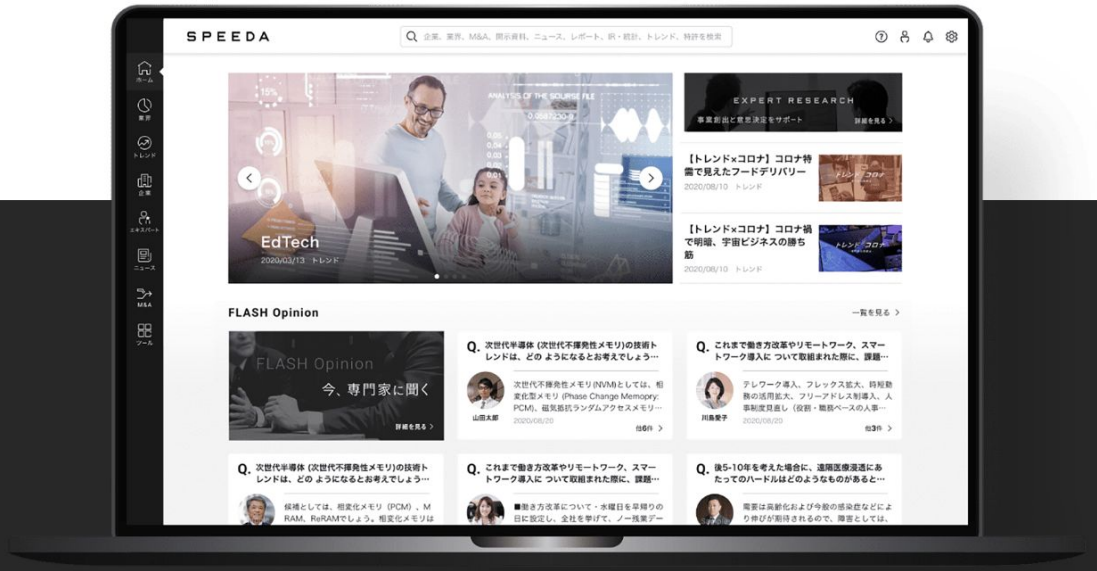
- 大学院〜ポスドク時代
  - 計算機代数学の研究
    - データサイエンスはほぼ無関係
    - プログラミングとアルゴリズムの修行の日々
- 前職時代
  - アウトソース型のデータサイエンティスト
    - 顧客の課題を機械学習を使って解く (PoC)
    - 数十種類の業種業務のデータに触れた



「自社のデータで分析したい」

UZABASE のサービス SPEEDA の紹介

SPEEDA



市場分析や競合調査、
経営の意思決定を支える
経済情報プラットフォーム。

業界分類

560 業界

業界レポート	ビジネスレポート	上場・非上場企業	スタートアップ	M&A案件
3,000 部以上	2,000 媒体	950 万 社以上	15.2 万 社以上	195 万 件

IR・統計	トレンド	登録エキスパート	特許動向
10 万 統計	90 部以上	8,000 人	332 分類

詳しくはこちら →

https://jp.ub-speeda.com/

データサイエンス取り組み例

企業の業界推定

- 企業情報の事業内容などのテキストから、SPEEDA の業界をタグ付けする
- アナリストが予めタグ付けた教師データを用いるクラス分類（文書分類）

ニュースの企業紐付け

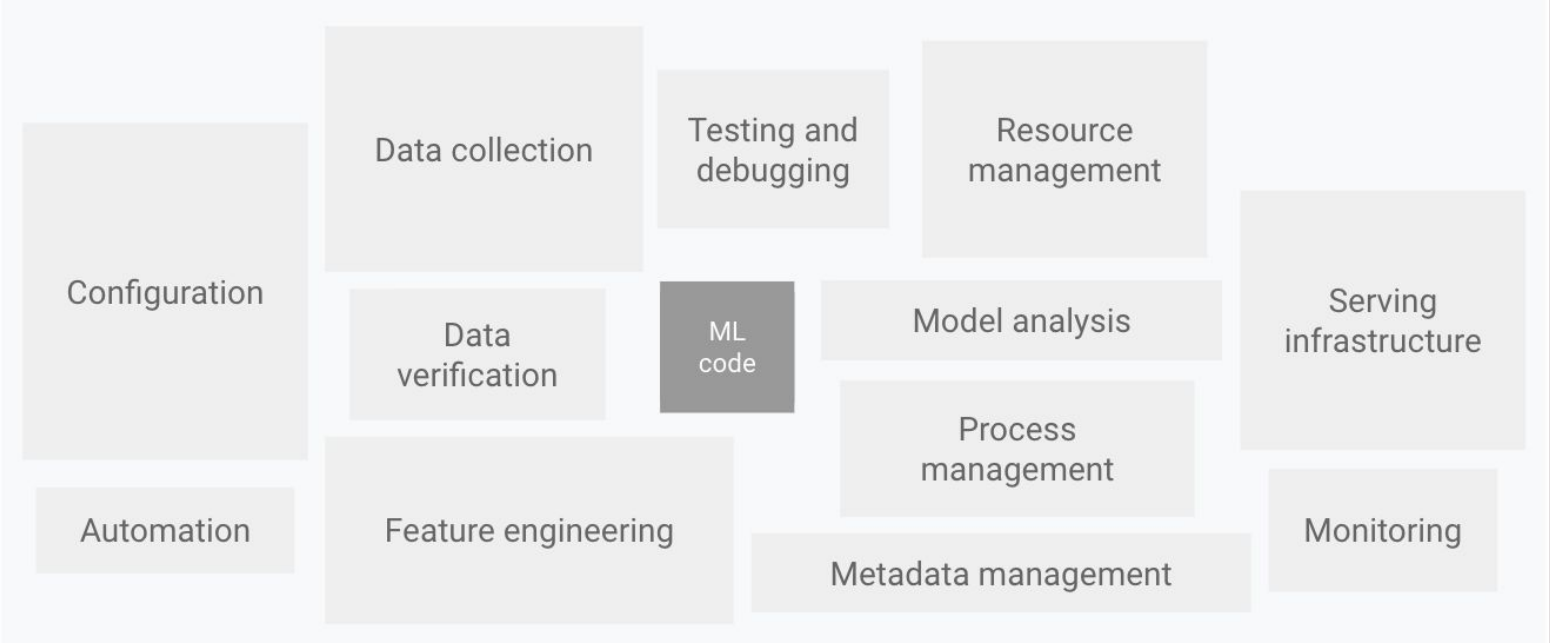
- ニュース（毎日数万件）の文章中で言及されている企業を抽出
- 社員がつけた企業タグを教師データにした固有表現抽出

B2B SaaS事業のデータサイエンティスト

私たちは機械学習モデルをプロダクトとしてデプロイし、
ユーザに価値を届けることに重点を置いています。

モデルのチューニングによって精度を追い求めるだけではなく、
データの蓄積から学習パイプラインや精度改善のループ構築、
予測APIの提供までの全てを一気通貫で設計、実装しています。

これまで: 構成管理



https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

機械学習のコードだけでは価値を届けられない

想定ケース

データを取得して、
教師あり学習し、
できたモデルをAPIとしてサーブする。

使用技術

PyTorch Lightning

scikit learn

で学習したモデルを、

FastAPI

でAPI化するコードを、

docker

でコンテナ化し、

kubernetes

環境で動かす、

ANSIBLE

で構成管理する、

Jenkins

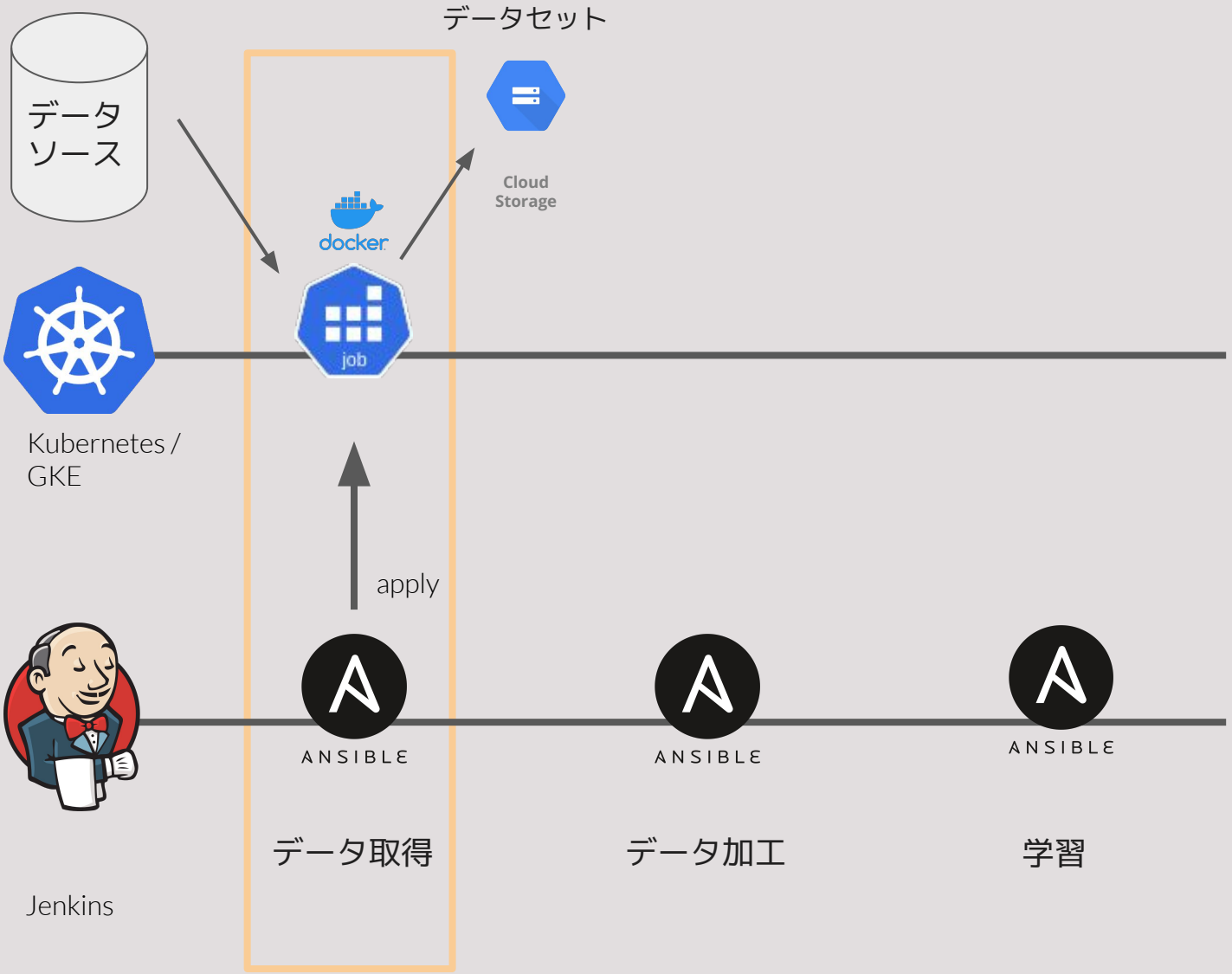
パイプライン。

パイプライン（学習）

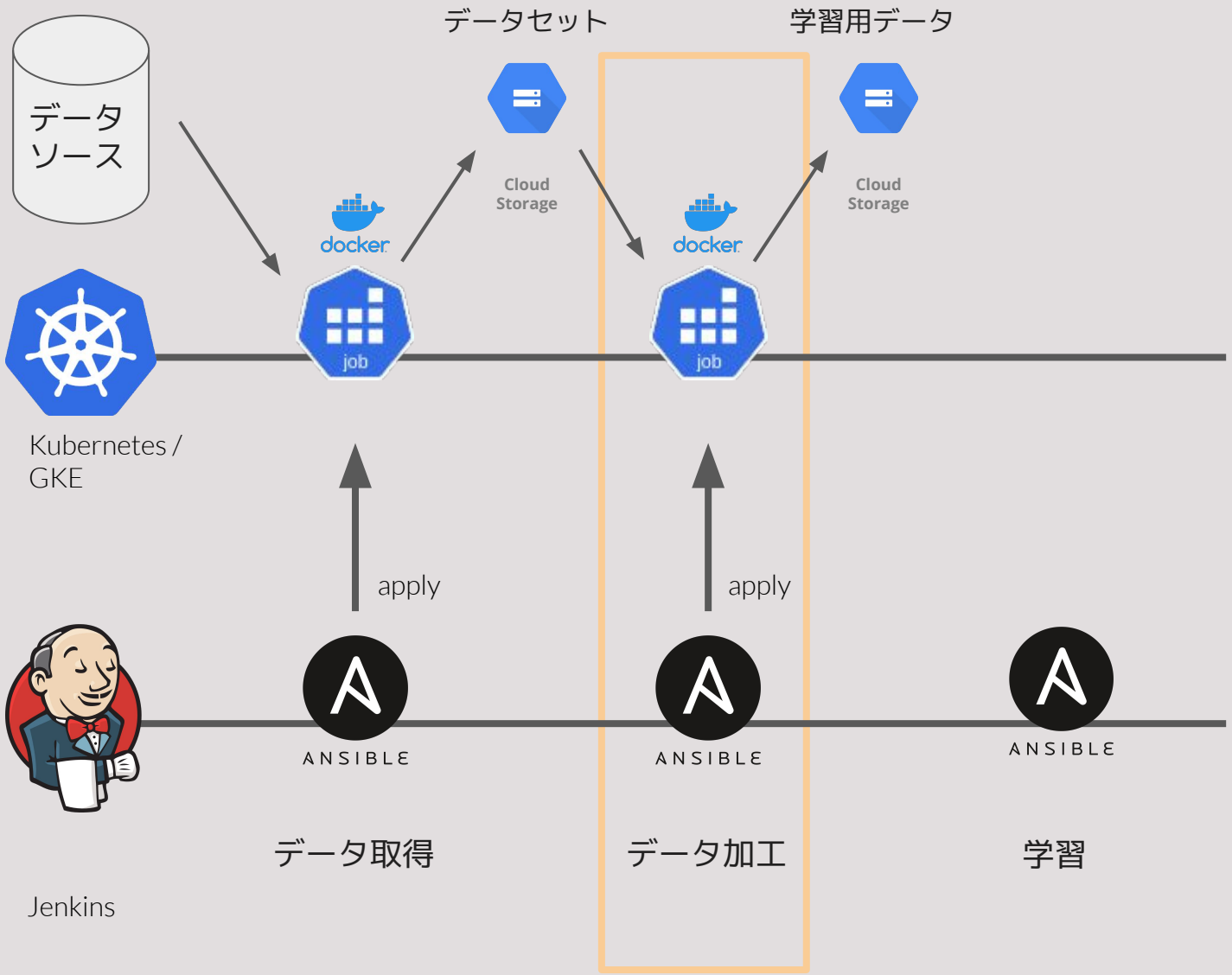
データ取得　　データ加工　　学習

Jenkins

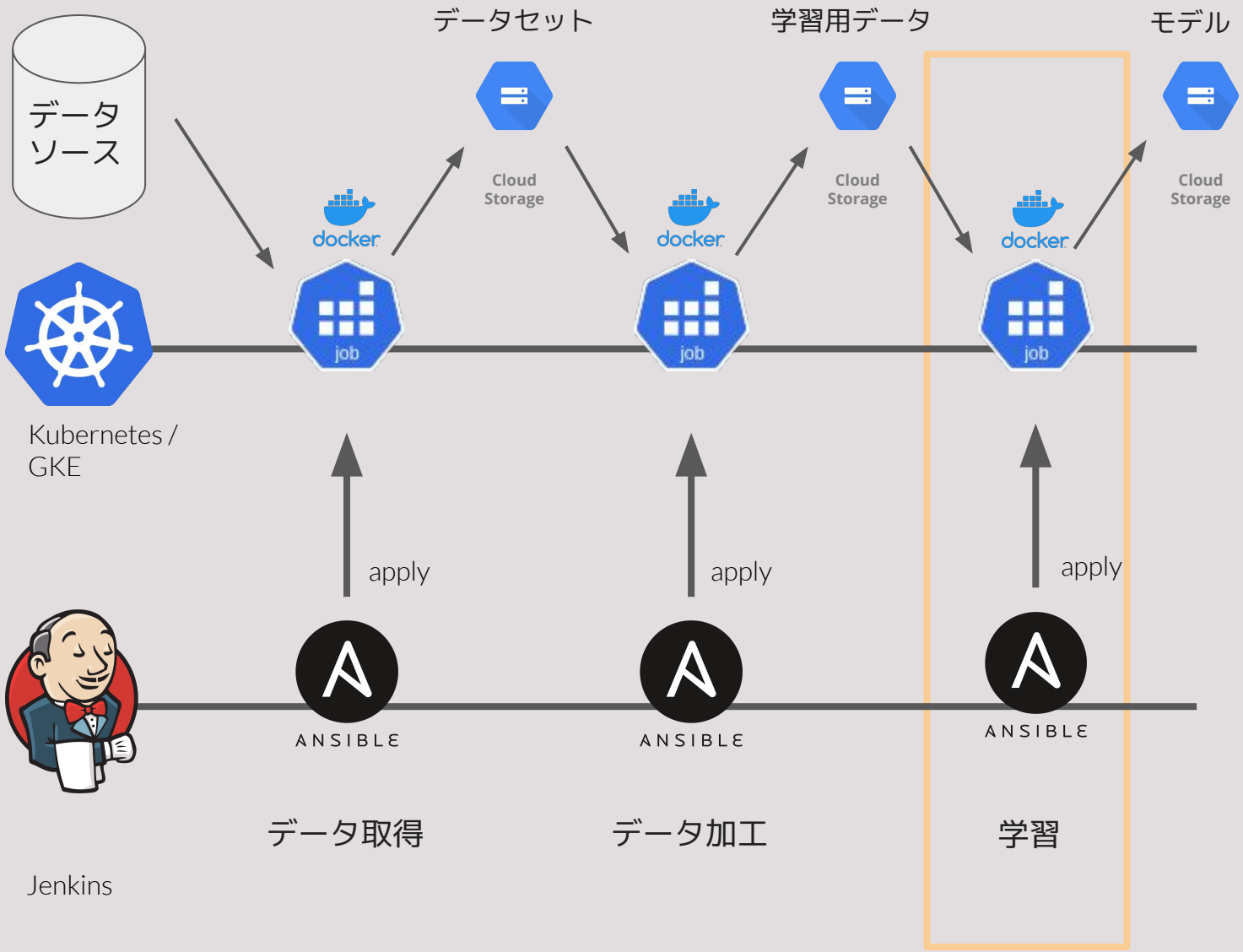
パイプライン（学習）


データセット
データ
ソース
Cloud
Storage
docker
job
Kubernetes /
GKE
apply
ANSIBLE
ANSIBLE
ANSIBLE
データ取得
データ加工
学習
Jenkins

パイプライン（学習）


データセット
学習用データ
データ
ソース
Cloud
Storage
Cloud
Storage
docker
docker
job
job
Kubernetes /
GKE
apply
apply
ANSIBLE
ANSIBLE
ANSIBLE
データ取得
データ加工
学習
Jenkins

パイプライン（学習）


データセット
学習用データ
モデル
データソース
Cloud Storage
Cloud Storage
Cloud Storage
docker
docker
docker
job
job
job
Kubernetes / GKE
apply
apply
apply
ANSIBLE
ANSIBLE
ANSIBLE
データ取得
データ加工
学習
Jenkins

何ができるようになったか

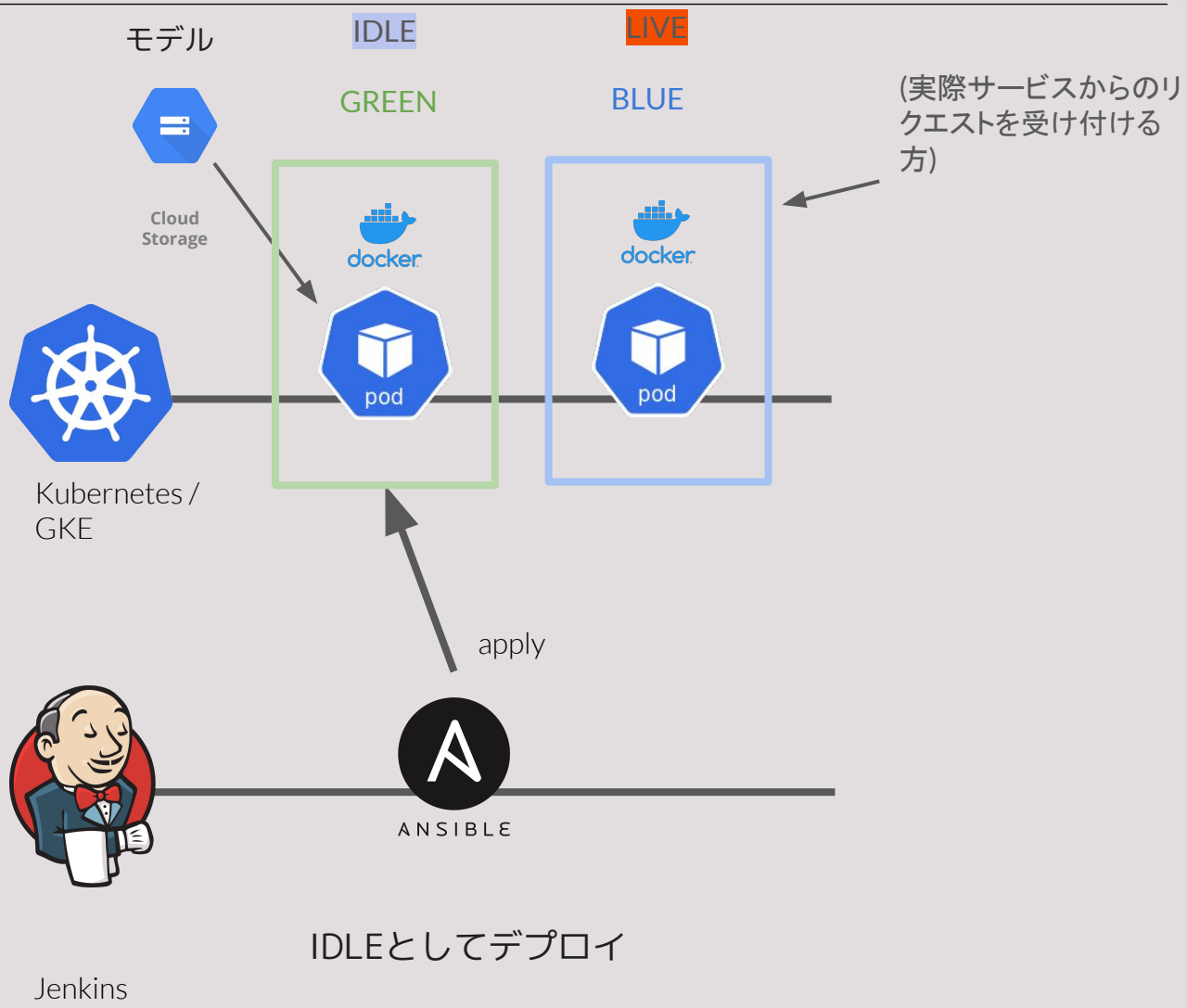
ボタンひとつで、
最新のデータで学習したモデルができる。

データ取得、データ加工、学習
ステップ毎の成果物に日付を付与してバージョニングすることによって、
再現性を確保している。
問題の切り分けが簡単になった。

ステップ単体の実行も可能。

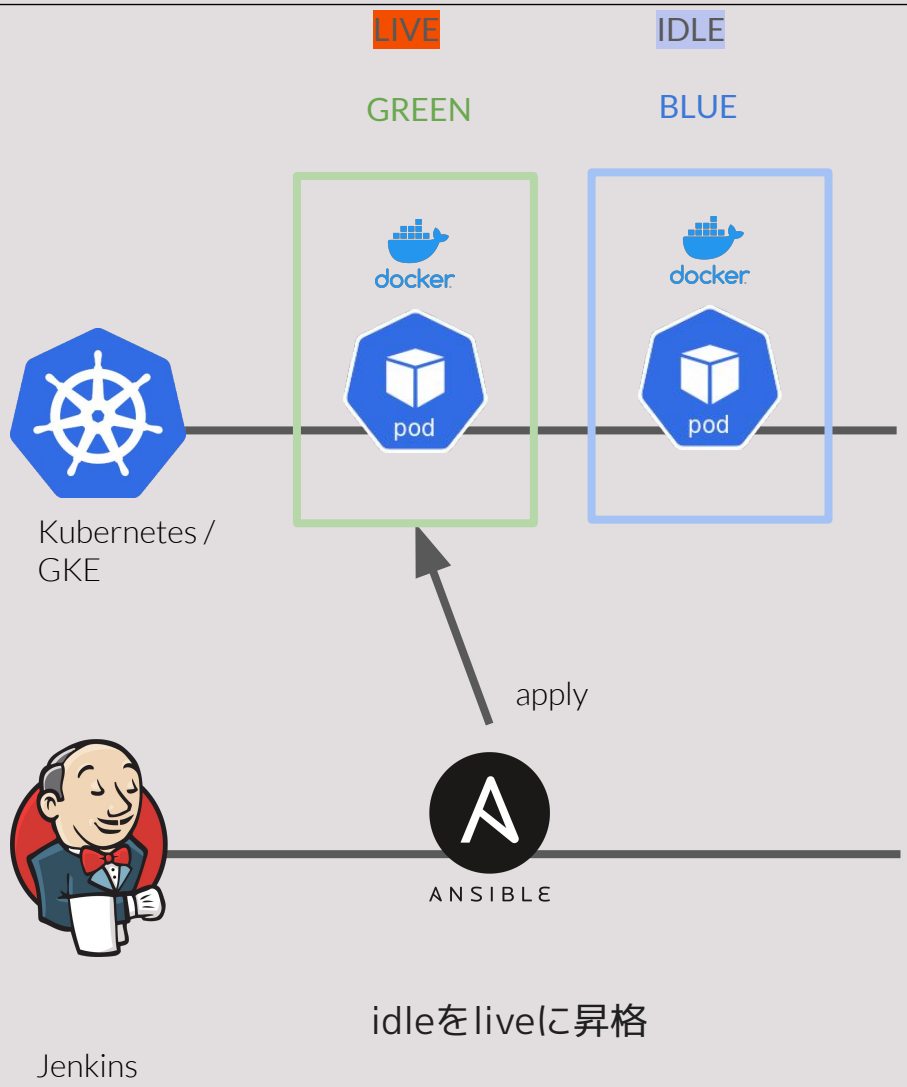
パイプライン（API）

Istioを用いた
Blue Green Deployment



パイプライン（API）

Istioを用いた
Blue Green Deployment



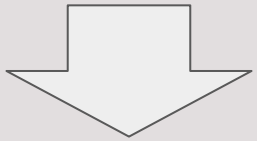
何ができるようになったか

ボタンひとつで、
最新のモデルをデプロイ。

問題発生時、すぐ切り戻せる。
モデルのバージョンを指定することによって、
特定のモデルをデプロイできる。

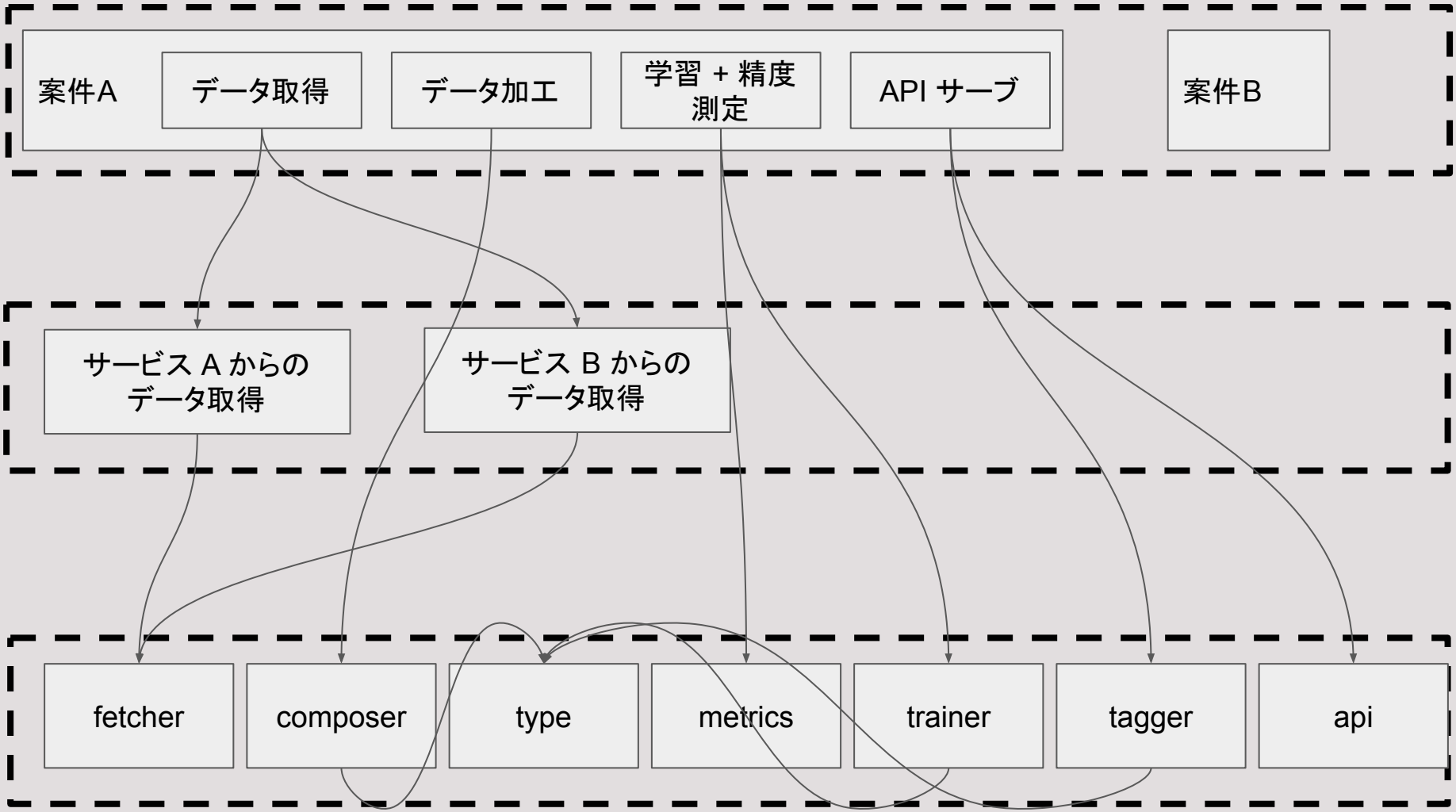
これまで: Python コンポーネントづくり

- 事業の成長と共に増えるデータ（量・種類）
- 種類が一つ増えると、既存データとの組み合わせで紐付けニーズが発生する
- 再利用可能な処理がとても多い
  - 学習、精度評価、データの保存、、、
  - 「処理は共通、データが違えば違う問題が解ける」



Python の再利用可能な基本コンポーネント集を作り続けている

コンポーネントたち (概要)



コンポーネント群の基本設計

- 大きく 3 レイヤに分かれている
  - 案件レイヤ: 案件特有の処理、設定
  - データカタログレイヤ: 各サービスからのデータ取得処理
  - 基本部品レイヤ: 案件にもサービスにも関係ない処理
- レイヤまたいで逆流した依存の禁止
  - 依存パッケージ記述として一方通行
  - 概念としても一方通行

コンポーネント群をチームで活用

- 精度向上の試行錯誤でも活用
  - 共通処理を何度も書かない
  - パイプライン化を容易にする
- 見積もりやチーム共通語彙としても使う
  - trainer 一つの開発は xx ポイント

何ができるようになったか

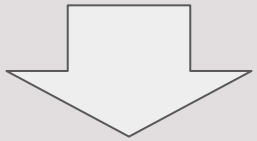
全体的な生産性の向上

チーム開発も機能してきた

案件こなすごとに共通の部品を拡張し続ける。
モデルの試行錯誤にも使うことで、パイプライン化も容易に。
チーム内で共通語彙化。

これまで: 機械学習周辺の話

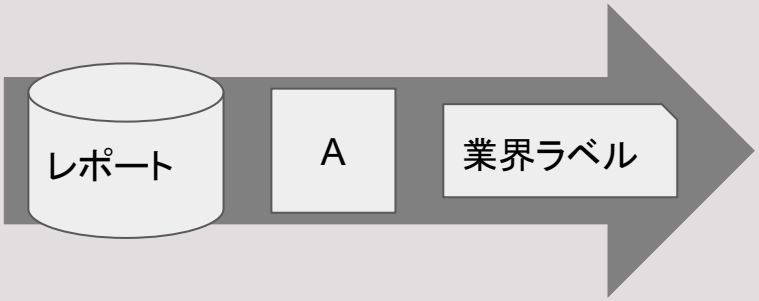
- 教師データの確保が課題
  - 紐付けたい = 紐付いていない
  - データの専門性も高く、外部リソースに出しづらい
- 各コンテンツの量は豊富
  - 業界レポートはラベルついたテキスト

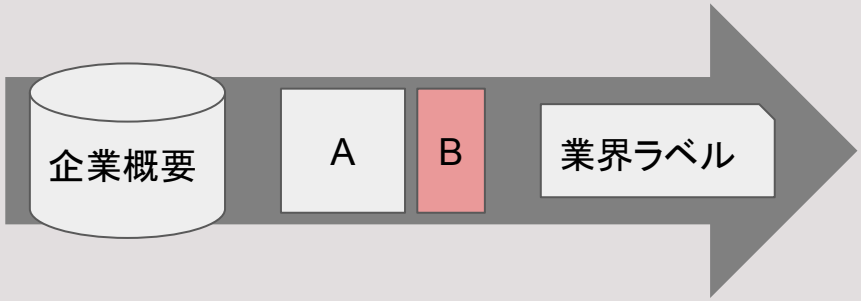


コンテンツを利用して事前学習モデルを作る

コンテンツを使った事前学習モデルの生成

- ラベルに関連したコンテンツを分類するモデルを作る
  - 例: 業界レポートで業界を分類 (A)
- 上記モデルにドメイン適応層を追加して、分類したいデータで再学習
  - 例: 業界レポート-業界モデルに全結合層 (B) 追加して、企業概要で業界を分類
  - (A) の学習率は少し弱める
- (A) は他の課題にも再利用できる





何ができるようになったか

コンテンツをうまく利用したモデル作り

モデル同士の再利用

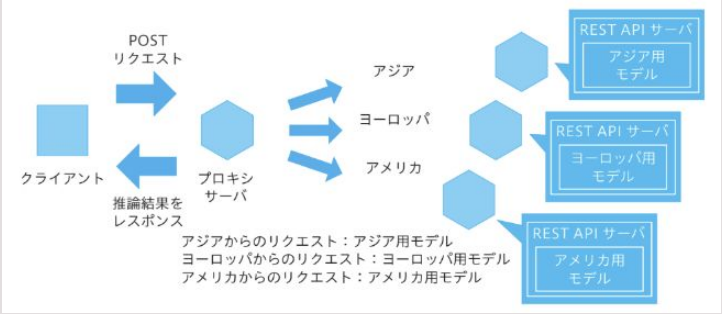
自社特有の資産（コンテンツ）をうまく使って、
教師データ不足もカバー

これまで: まとめ

- 構成管理
  - データ更新に追従して学習
  - 最新モデルのデプロイ/切り戻しの自動化
- Python コンポーネントづくり
  - 共通化で生産性アップ
  - チーム開発化
- 機械学習周辺の話
  - コンテンツで事前学習して精度向上
  - 社内事前学習モデル集

これから: 推論結果の小さなリリース

- モデルのスコアだけが正義ではない
  - 顧客ごとに、重要視するデータは異なる
    全体的に改善しても、一部の顧客には改悪となるケースがある
- 使うモデルをデータで分ける
  - すでに地域別や言語別は導入している
- モデル刷新時に、一部のデータにだけ先行適用
  - 顧客へのインパクトを抑えつつ徐々に新しいモデルに移行していく
  - 推論結果の管理コストは最小限に



「AIエンジニアのための機械学習システムデザインパターン」より引用

これから: データメンテナンスのためのデータサイエンス

- 誤ったラベルの検出のためのデータサイエンス
  - 教師データも変化する = 時間と共にラベルが間違ったことになる可能性
  - 教師ラベルとして他と傾向の異なるものの検出
- 効果的な教師データ作成のためのデータサイエンス
  - 能動学習
- タグ体系のメンテナンスのためのデータサイエンス
  - 2 つの近しいタグ体系の統合
  - タグ体系内のタグの統合/廃止/新設

まとめ

- これまで
  - 構成管理
    - ML モデルの管理コスト低減
  - Python コンポーネントづくり
    - チーム開発の生産性向上
  - 機械学習周辺の話
    - モデル作りの生産性向上
- これから
  - 推論結果の小さなリリース
    - 事業貢献をよりタイムリーに
  - データメンテナンスのためのデータサイエンス
    - より長期的にデータサイエンスやりやすくしていく