



FOR HEALTHCARE
AND LIFE SCIENCES

最新事例に学ぶ創薬研究領域のAWSクラウド活用セミナー2023

AWSを活用したMLOpsの加速

原田 裕平

アマゾンウェブサービスジャパン合同会社
ソリューション アーキテクト

自己紹介

原田 裕平(Harada Yuhei)

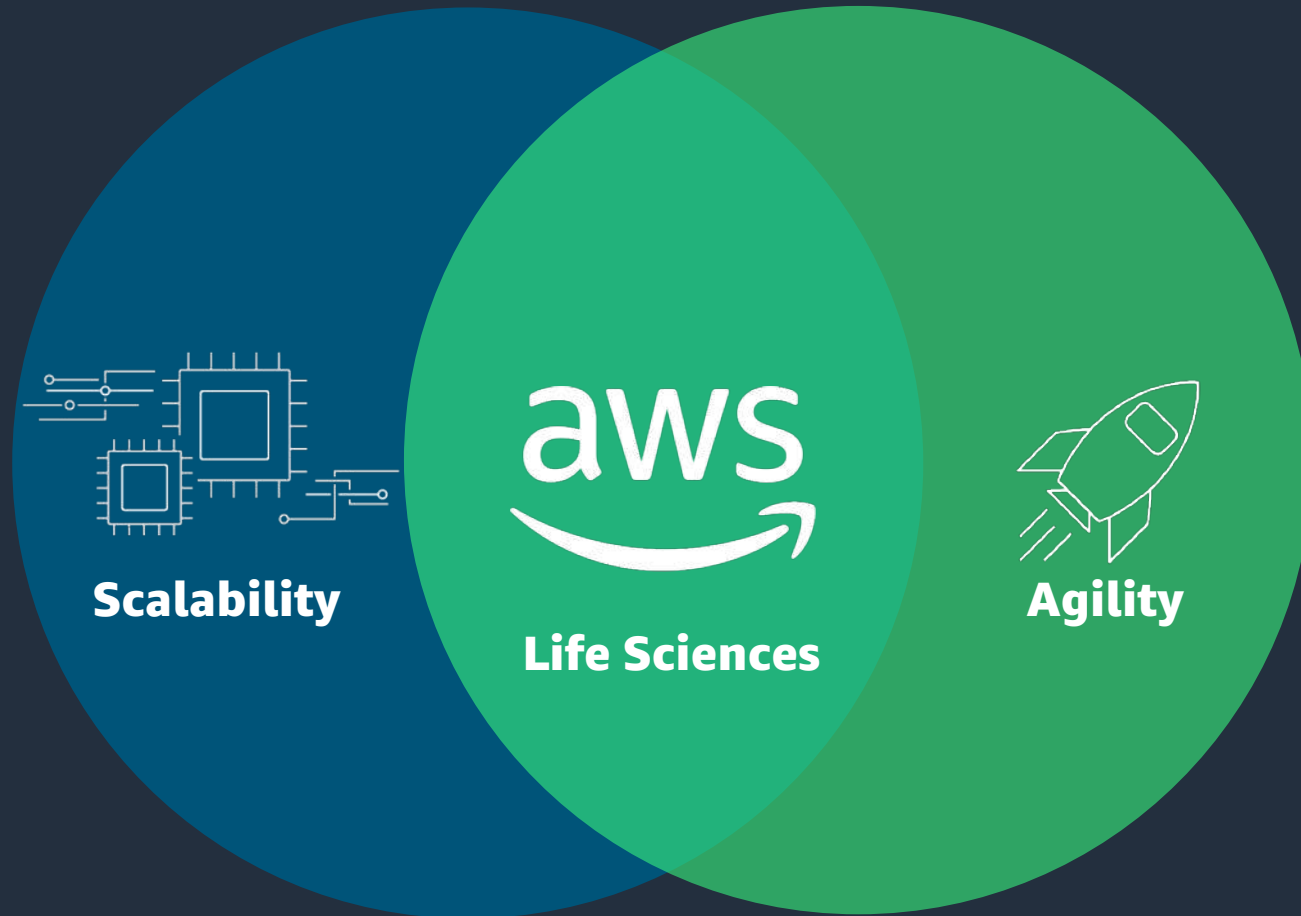
yuhehara@amazon.co.jp

アマゾンウェブサービスジャパン合同会社
ソリューション アーキテクト

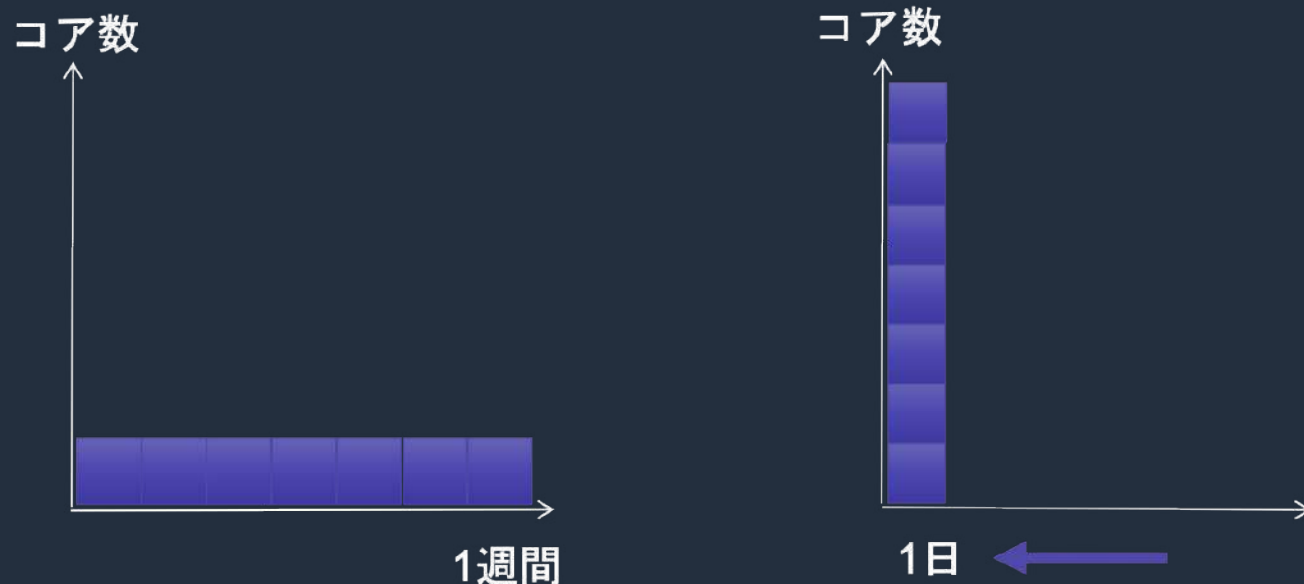
製薬業界のお客様を中心に
AWS活用をご支援しています



創薬研究におけるAWSの提供する価値



スケーラビリティによりコスト効率よく研究のスループット向上



コンピューティング費用は「時間x台数」の積算なので
逐次処理をしても並列処理をしても費用は同じ



[Amazon SageMaker](#)



AWS Batch



AWS ParallelCluster

ビルディングブロックの活用で試行錯誤のサイクルを加速

AWSはやりたいことをご自身で実現するSelf Service Platform

- 1つのサービスやツールでは自由度と実装コストの両立に限界

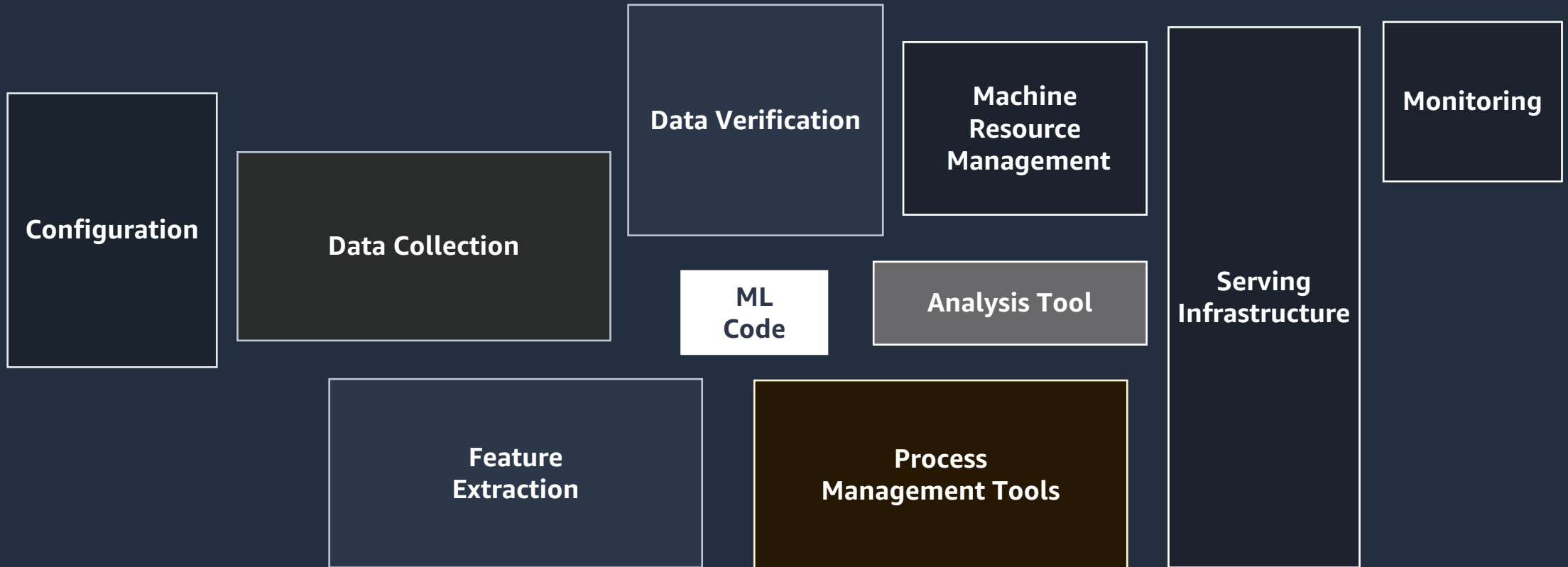
⇒ 複数のサービスを適材適所で組み合わせ、**やりたいことを最小の手間**で実現する

➡ **Building Block**



- 価値創出にフォーカス
- 失敗や試行錯誤が容易
- リードタイムの短縮

機械学習を用いた課題解決に必要な要素



“Only a small fraction of real-world ML systems is composed of the ML code”

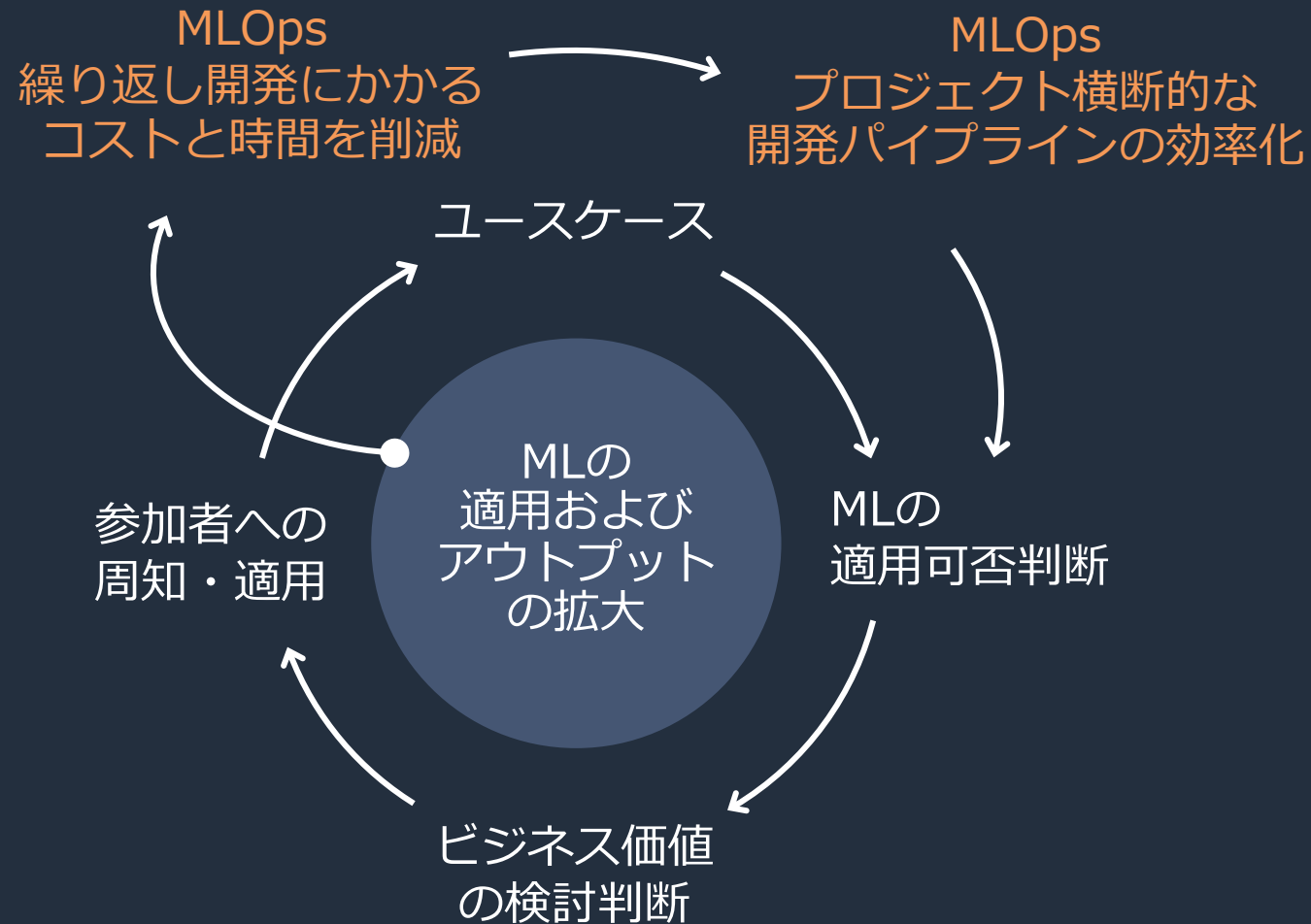
source: Hidden Technical Debt in Machine Learning Systems [D. Sculley, & al.] – 2015

<https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf>

AWSにおける機械学習フライホイール

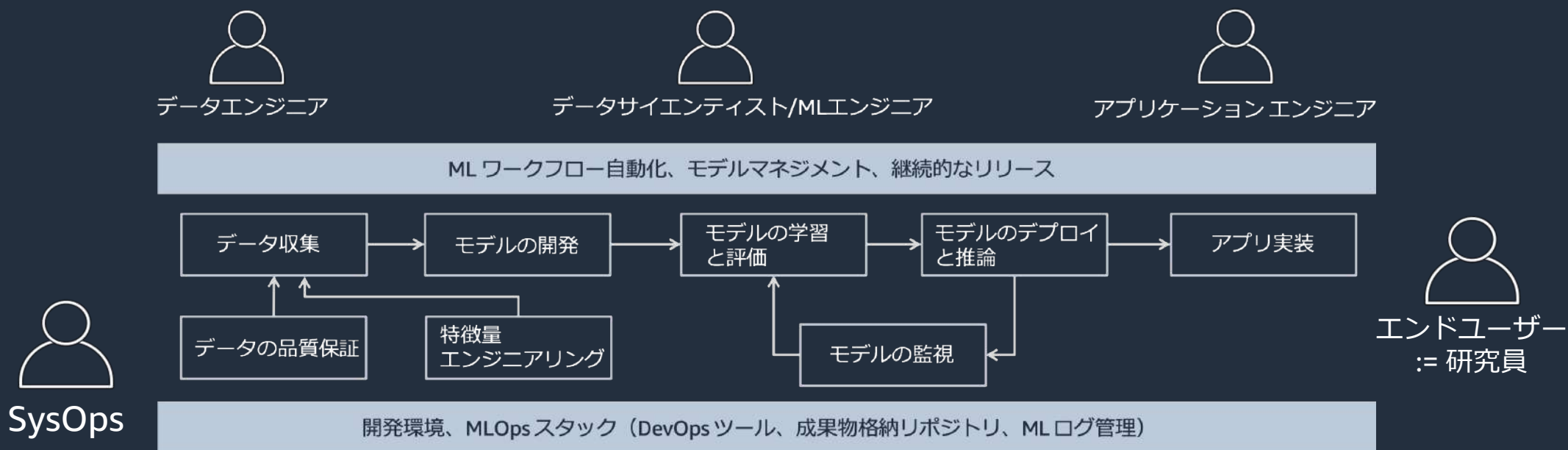


AWSにおける機械学習フライホイール



創薬研究における機械学習(ML)の課題とMLOps

1. MLの手法そのものが研究対象であり、開発には高度な専門性とドメイン知識が求められることから、**小規模チームでのMLOpsの実現が求められる**
2. MLアプリケーションの利用者もまた別の研究員であり、**試行錯誤、開発、改善ライフサイクルが短い**



創薬研究における機械学習(ML)の課題とMLOps

1. MLの手法そのものが研究対象であり、開発には高度な専門性とドメイン知識が求められることから、**小規模チームでのMLOpsの実現が求められる**
2. MLアプリケーションの利用者もまた別の研究員であり、**試行錯誤、開発、改善ライフサイクルが短い**

研究員が必要かつ重要な作業にフォーカスできる
環境およびメカニズムの構築が必要

SysOps

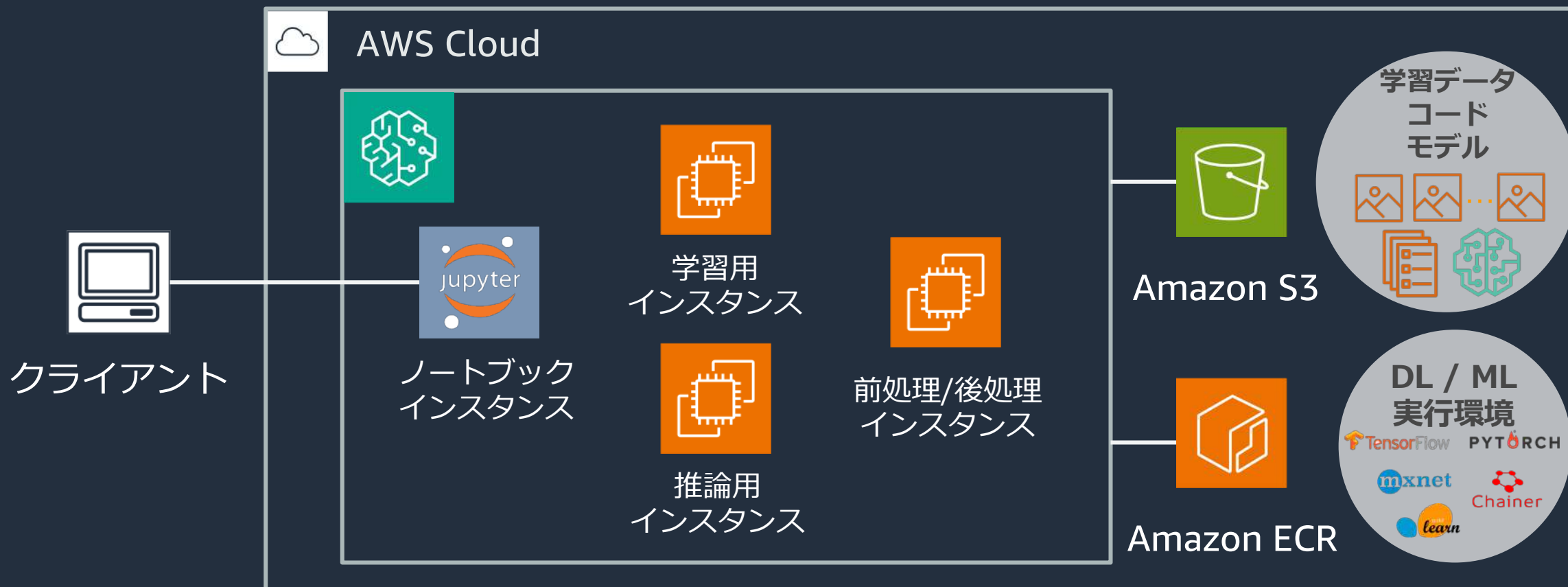
開発環境、MLOps スタック (DevOps ツール、成果物格納リポジトリ、ML ログ管理)

Undifferentiated Heavy Lifting

by Jeff Bezos, Founder and CEO of Amazon.com

他との差別化になりづらい重労働

Amazon SageMakerの基本アーキテクチャ



Amazon SageMakerの基本アーキテクチャ



クラ

チームで標準化された開発環境を迅速に構築
適切なコンピューティングリソースで試行錯誤
学習/推論/前・後処理を簡単にスケールアップ/スケールアウト

AWSでのワークフロー管理ツールの選択肢



AWS Step Functions

- サーバーレスオーケストレーションサービスを提供するサービス
- 様々なAWSサービスとの連携が用意
- 定義したステートマシンはAWS コンソールから「ワークフロー」という形式で見やすく可視化できる
- ステートマシンの各ステップの実行履歴をログから追跡できる



Amazon SageMaker Pipelines

- 機械学習ワークロードの CI/CD を実現する Amazon SageMaker の機能
- SageMaker Studio上でGUIでの開発が可能
- 各ステップの処理結果は SageMaker Experiments で記録され、モデルの出来映えや学習パラメータなどを視覚化できる

AWS Step Functions

ASL(Amazon States Language)と呼ばれるJSON形式の言語でワークフローを定義

ステートマシンの定義

Amazon ステート言語 (ASL) を使用してステートマシンを定義し、ワークフローのビジュアル表現を確認します。 [詳細はこちら](#)

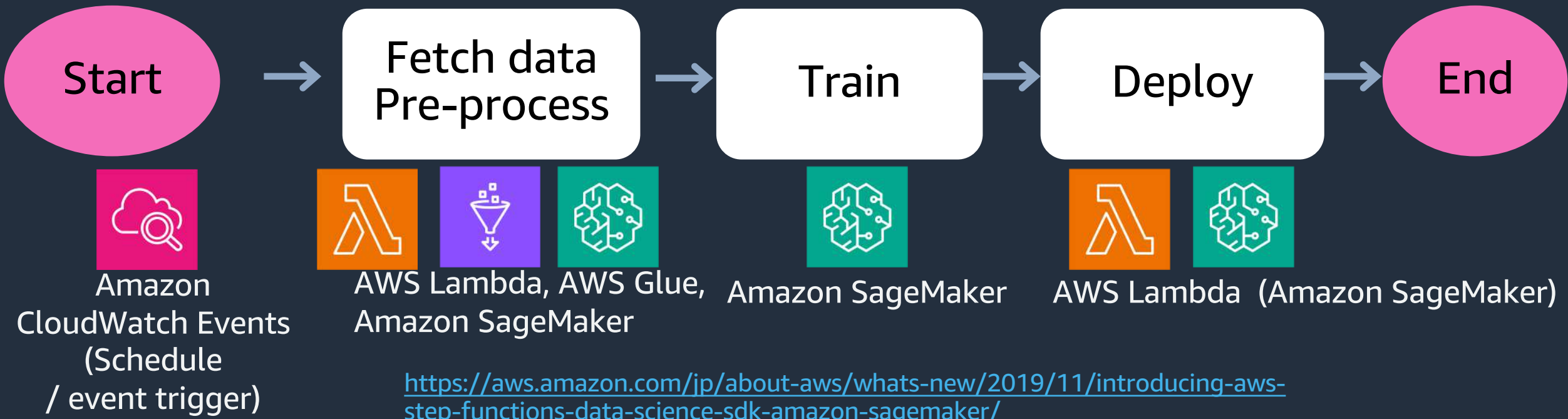
コードスニペットの生成 [詳細はこちら](#)

```
1 * [
2   "Comment": "An example of the Amazon States Language using a
3   parallel state to execute two branches at the same time.",
4   "StartAt": "Parallel",
5   "States": {
6     "Parallel": {
7       "Type": "Parallel",
8       "Next": "Final State",
9       "Branches": [
10      {
11        "StartAt": "Wait 20s",
12        "States": {
13          "Wait 20s": {
14            "Type": "Wait",
15            "Seconds": 20,
16            "End": true
17          }
18        }
19      }
20    ]
21  }
22 }
```

```
graph TD
  Start((Start)) --> Parallel
  subgraph Parallel
    direction LR
    Wait20s[Wait 20s]
    Pass[Pass]
  end
  Parallel --> Wait10s[Wait 10s]
  Wait10s --> FinalState[Final State]
  FinalState --> End((End))
```

AWS Step Functions を使ったMLワークフロー

- AWS Lambda, Glue などにも対応したサーバーレスオーケストレーション
- **AWS Step Functions Data Science SDK** を使って、Pythonで 前処理 → 学習 → デプロイ のワークフローを作成し、可視化



AWS Step Functions を使ったMLワークフローの構築

AWS Step Functions Data Science SDK

AWS Step Functions Workflow Studio

```
Create the TrainingStep for the Workflow

training_step = steps.TrainingStep(
    "SageMaker Training Step",
    estimator=sklearn,
    data={"train": sagemaker.s3_input(preprocessed_training_data, content_type="csv")},
    job_name="sagemaker-training-job",
    wait_for_completion=True
)
```

各ステップを数行のPythonコードで記述

```
Create and execute the Workflow

workflow_graph = Chain([processing_step, training_step, processing_evaluation_step])
branching_workflow = Workflow(
    name="SageMakerProcessingWorkflow",
    definition=workflow_graph,
    role=workflow_execution_role,
)

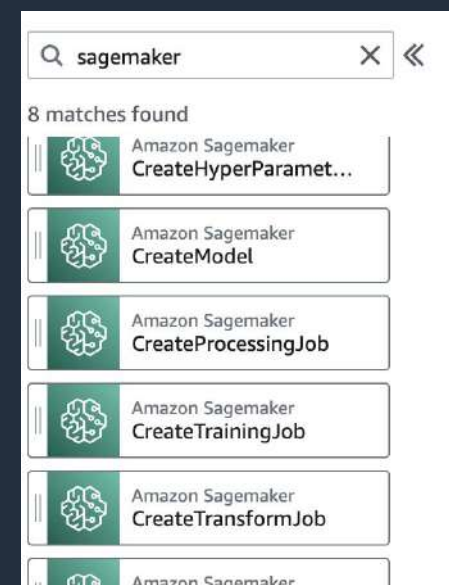
branching_workflow.create()

# Execute workflow
execution = branching_workflow.execute(
    inputs={"PreprocessedTrainingData": sagemaker.s3_input(preprocessed_training_data, content_type="csv")},
    execution_role_name="SageMakerProcessingWorkflowExecutionRole",
)

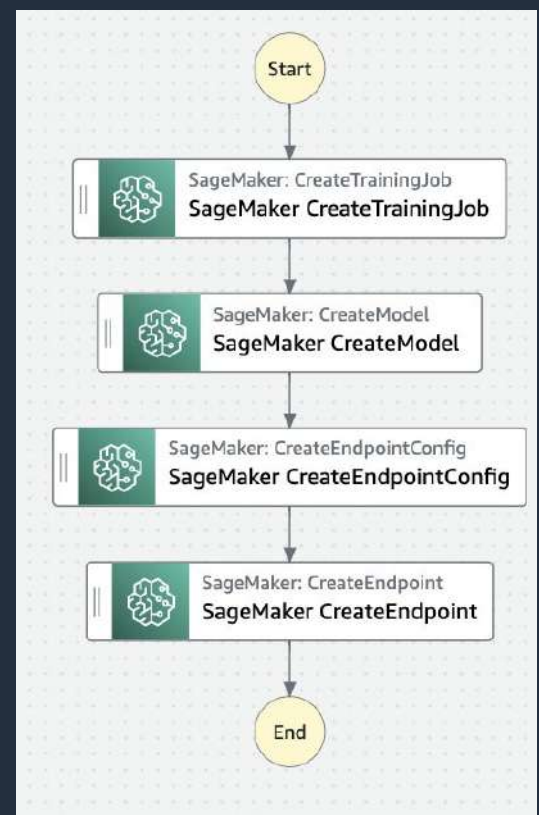
# Wait for the workflow to complete
execution.wait_for_completion()
```

ステップを Chain させて Workflow を定義し、作成、実行を行う

<https://aws-step-functions-data-science-sdk.readthedocs.io/en/stable/>



<https://aws.amazon.com/jp/blogs/news/define-and-run-machine-learning-pipelines-on-step-functions-using-python-workflow-studio-or-states-language/>





FOR HEALTHCARE
AND LIFE SCIENCES

Thank you!

Yuhei Harada

yuhehara@amazon.co.jp