aws SUMMIT Switzerland

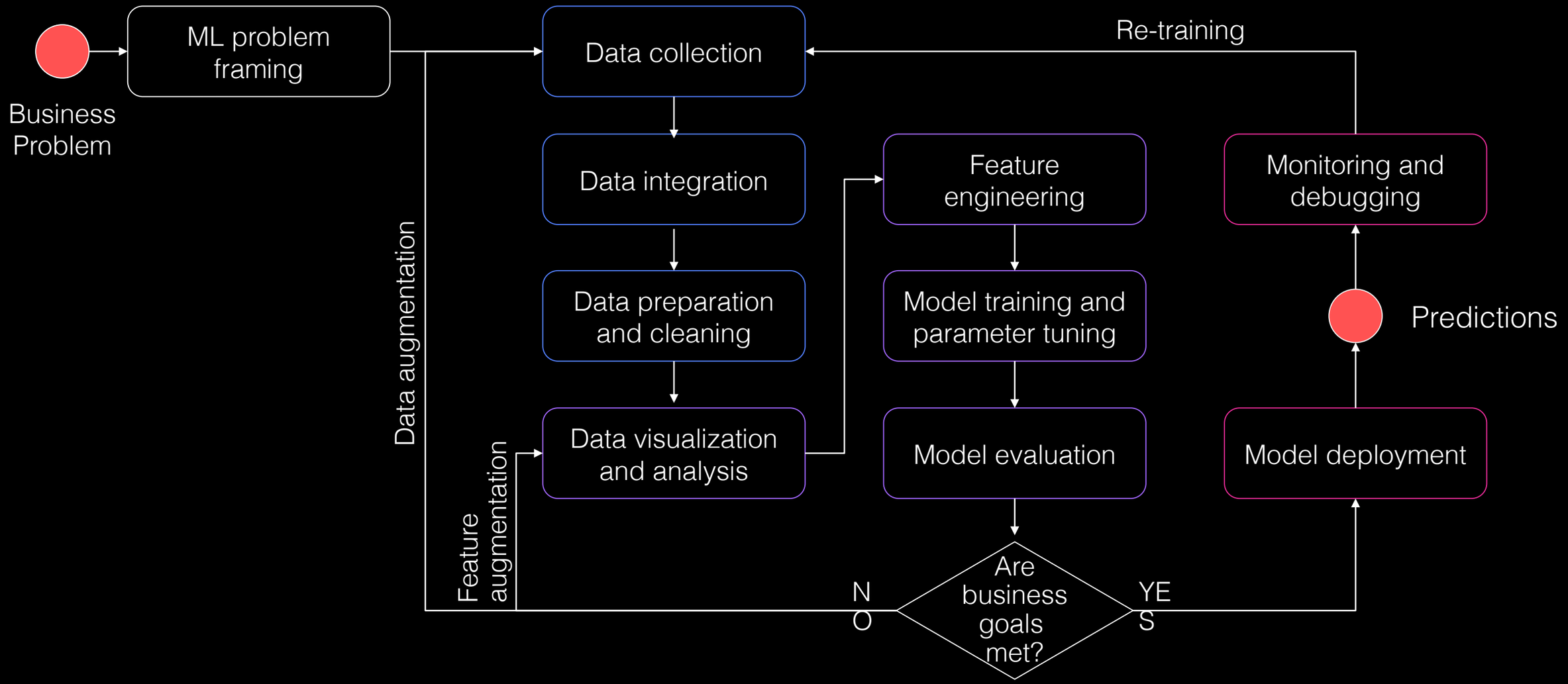
Build, Train and Deploy Machine Learning Models on Amazon SageMaker

Julien Simon
Global Evangelist, AI & Machine Learning
Amazon Web Services
@julsimon

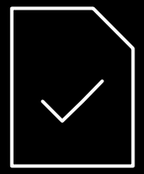
Stéphane Cheikh
Director, Portfolio Evolution using Artificial Intelligence
SITA

aws SUMMIT

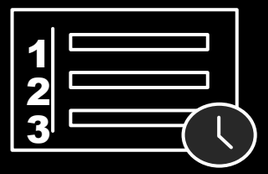
Machine learning cycle



Amazon SageMaker

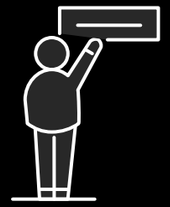


Collect and prepare training data



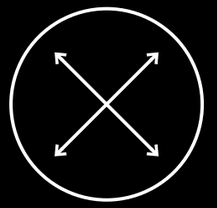
Choose and optimize your ML algorithm

Set up and manage environments for training



Train and tune ML models

Deploy models in production



Scale and manage the production environment

Same service and APIs from experimentation to production

intuit

tinder

KIA

CONVOY

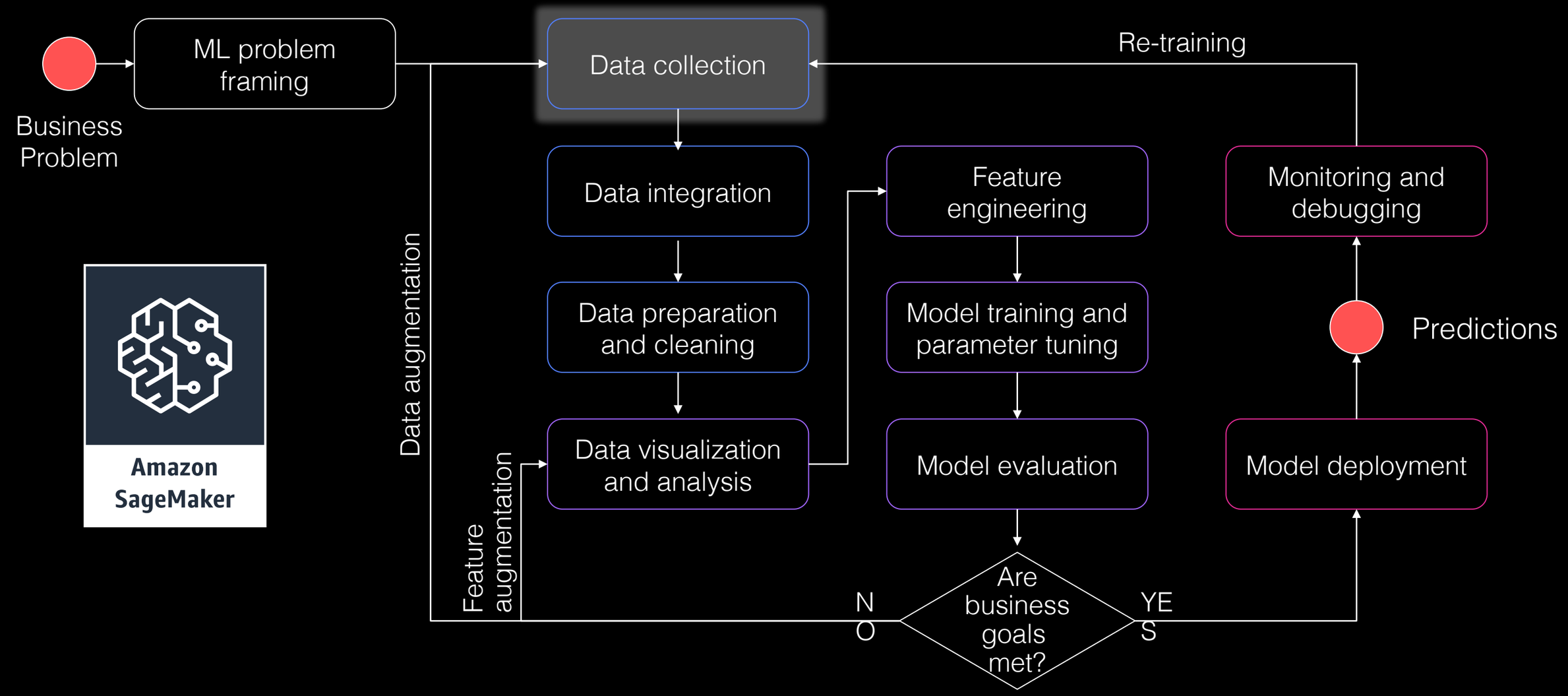
SIEMENS

DOW JONES

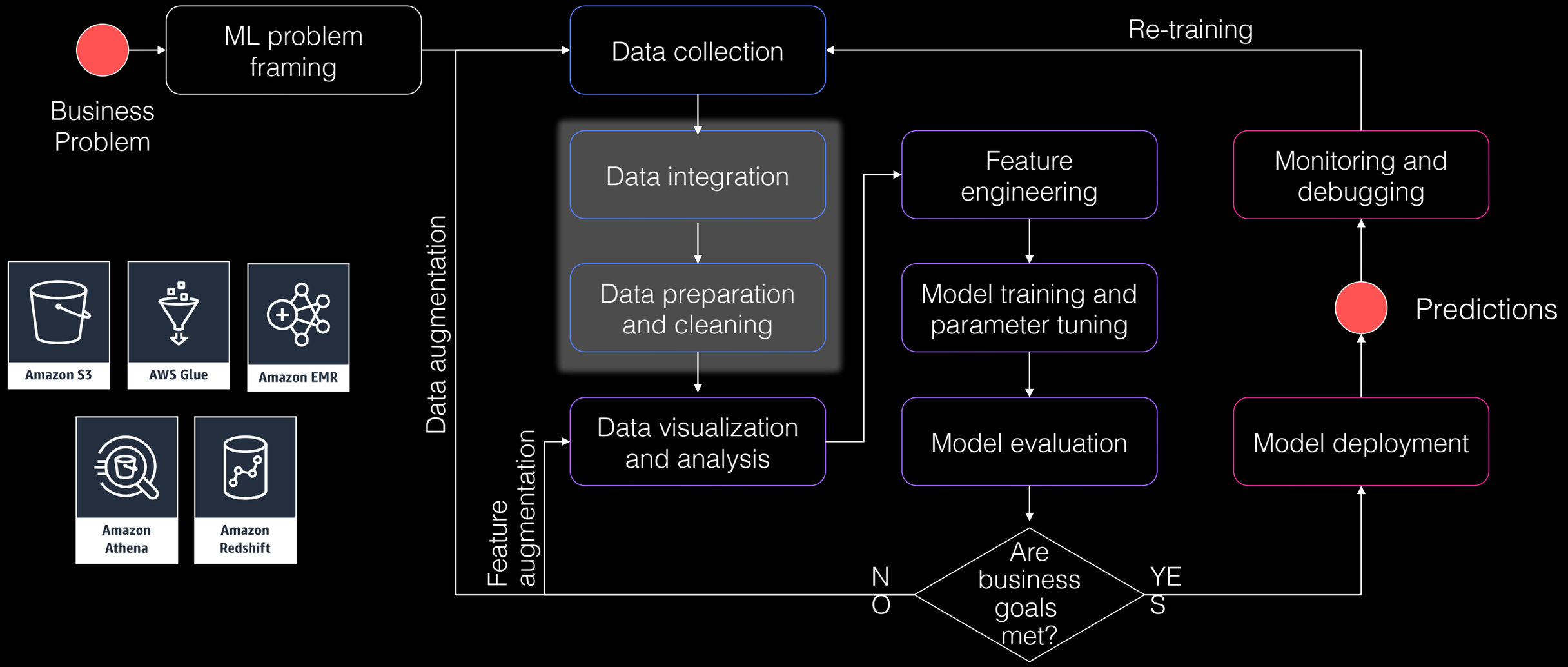
SONY

aws SUMMIT

Build your dataset

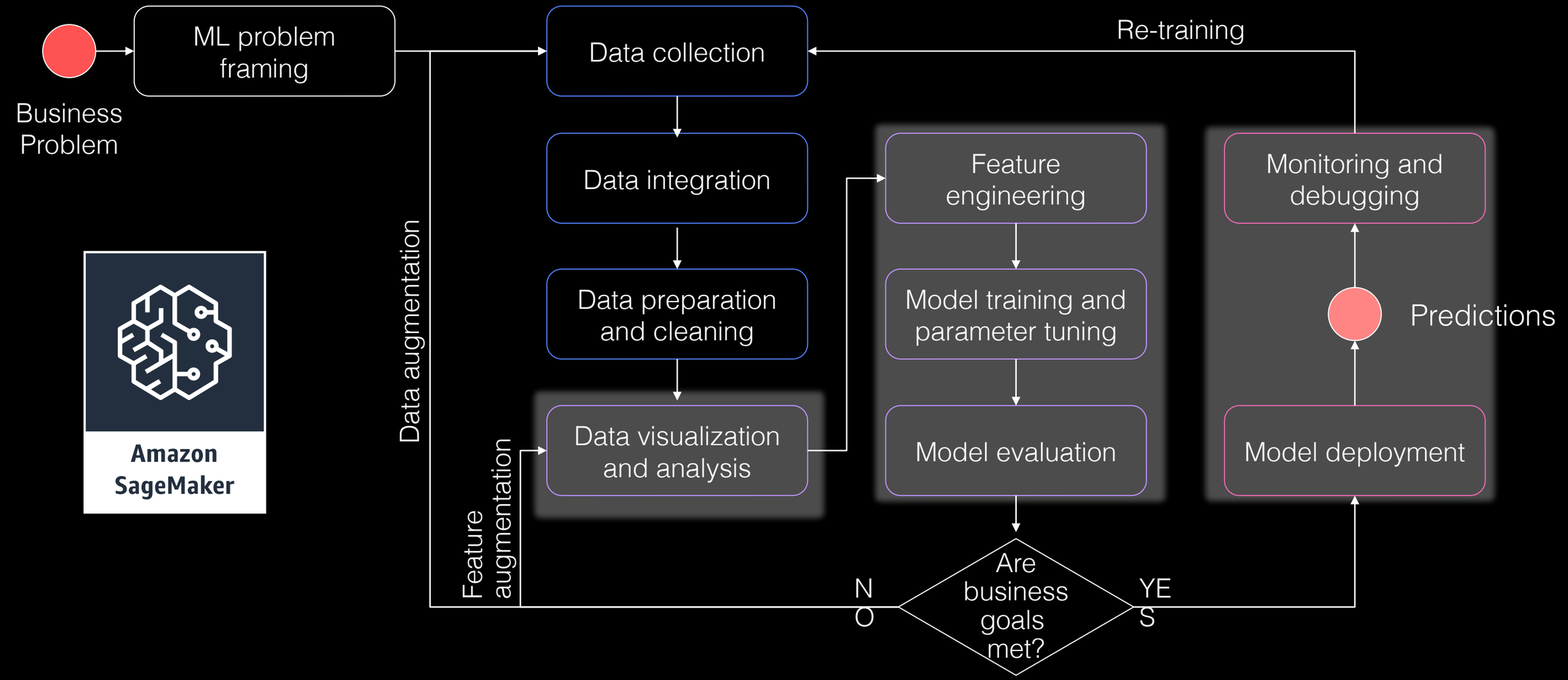


Prepare your dataset for Machine Learning



aws SUMMIT

Build, train and deploy models using SageMaker

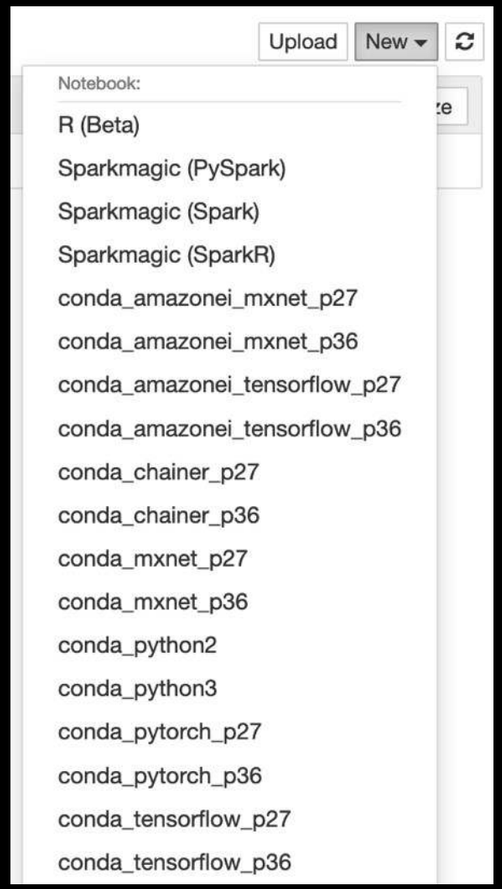


Building models with Amazon SageMaker

aws SUMMIT

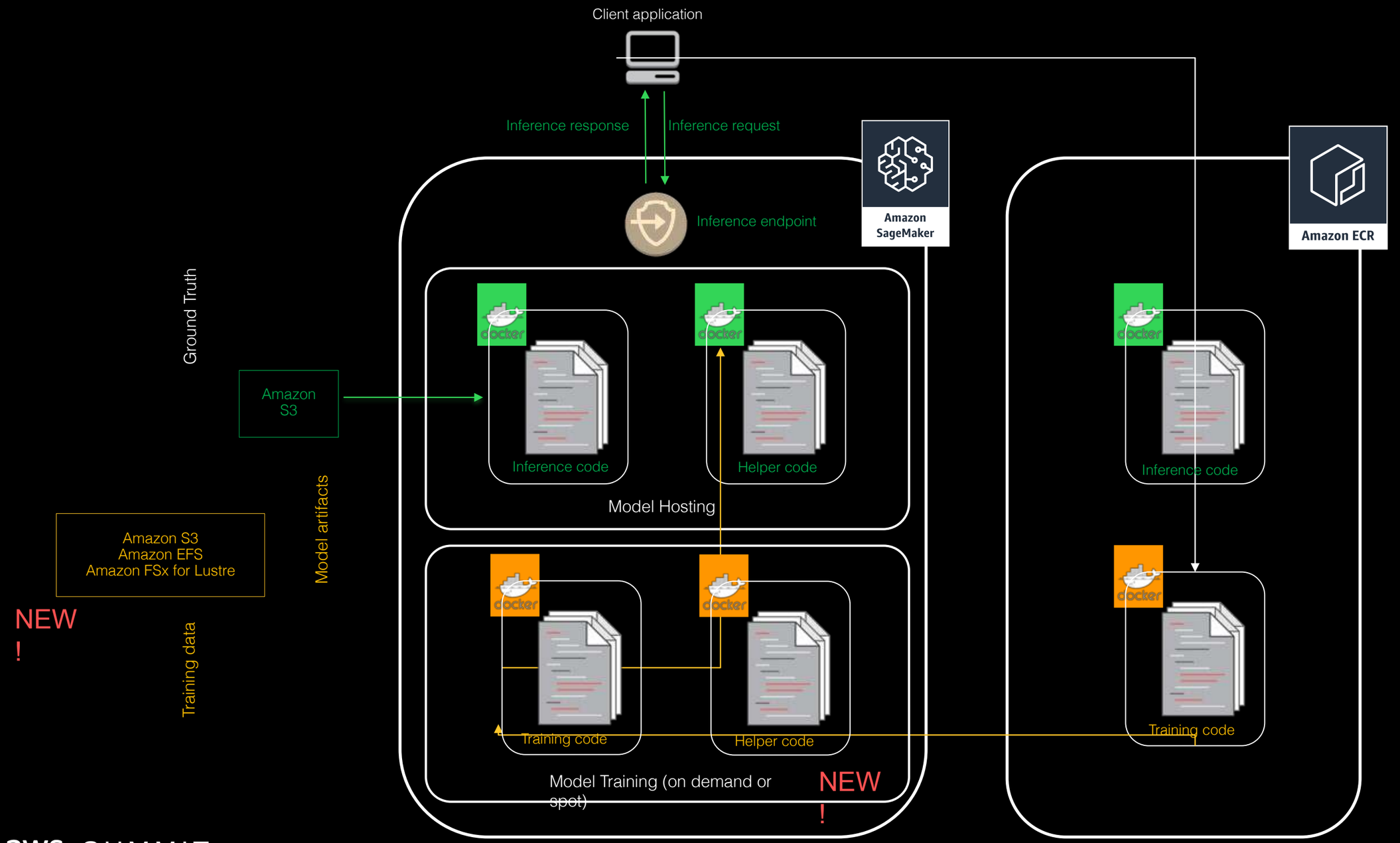
Notebook instances

- *Fully managed EC2 instances, from T2 to P3*
  - G4 and R5 now available for inference – NEW!
- Pre-installed with Jupyter and Conda environments
  - Python 2.7 & 3.6
  - Open-source libraries (TensorFlow, Apache MXNet, etc.)
  - Beta support for R – NEW!
  - Amazon Elastic Inference for cost-effective GPU acceleration
- Lifecycle configurations
- VPC, encryption, etc.
- Get to work in minutes, zero setup



The Amazon SageMaker API

- Python SDK orchestrating all Amazon SageMaker activity
  - High-level objects for algorithm selection, training, deploying, automatic model tuning, etc.
    https://github.com/aws/sagemaker-python-sdk
  - Spark SDK (Python & Scala)
    https://github.com/aws/sagemaker-spark/tree/master/sagemaker-spark-sdk
- AWS SDK
  - Service-level APIs for scripting and automation
  - CLI: '*aws sagemaker*'
  - Language SDKs: boto3, etc.



Model options

Training code

AWS Marketplace for Machine Learning: 250+ off-the-shelf algos and models

Factorization Machines
Linear Learner
Principal Component Analysis
K-Means Clustering
XGBoost
And more

Built-in Algorithms (17)

No ML coding required
No infrastructure work required
Distributed training
Pipe mode

mxnet Chainer TensorFlow PyTorch scikit-learn dmlc XGBoost

NEW !

Built-in Frameworks

Bring your own code: Script mode
Open-source containers
No infrastructure work required
Distributed training
Pipe mode

Bring Your Own Container

Full control, run anything!
R, C++, etc.
No infrastructure work required

aws SUMMIT

Built-in algorithms

Orange: supervised, yellow: unsupervised

Linear Learner: Regression, classification	**Image Classification**: Deep learning (ResNet)
Factorization Machines: Regression, classification, recommendation	**Object Detection (SSD)**: Deep learning (VGG or ResNet)
K-Nearest Neighbors: Non-parametric regression and classification	**Neural Topic Model**: Topic modeling
XGBoost: Regression, classification, ranking https://github.com/dmlc/xgboost	**Latent Dirichlet Allocation**: Topic modeling (mostly)
K-Means: Clustering	**BlazingText**: GPU-based Word2Vec, and text classification
Principal Component Analysis: Dimensionality reduction	**Sequence to Sequence**: Machine translation, speech to text and more
Random Cut Forest: Anomaly detection	**DeepAR**: Time-series forecasting (RNN)
Object2Vec: General-purpose embedding	**IP Insights**: Usage patterns for IP addresses
Semantic Segmentation: Deep learning	

Built-in frameworks: just add your code

- Built-in containers for training and prediction
  - Open-source, e.g., https://github.com/aws/sagemaker-tensorflow-containers
  - Build them, run them on your own machine, customize them, etc.

- Local mode: train and predict on your notebook instance, or on your local machine

- Script mode: migrate existing code to SageMaker with minimal changes

TensorFlow on AWS

C5 instances (Intel Skylake)

Training ResNet-50 with the ImageNet dataset using our optimized build of TensorFlow 1.11 on a c5.18xlarge instance type is designed to be 11x faster than training on the stock binaries

P3 instances (NVIDIA V100)

TensorFlow scaling efficiency with 256 GPUs

65 % Stock version

90 % AWS-optimized version

Apache MXNet: Deep learning for enterprise developers

Start with off-the-shelf models

- Gluon CV, Gluon NLP, Gluon TS
- ONNX compatibility

Fast and scalable training

- Keras-MXNet up to 2x faster than Keras-TensorFlow
- Near-linear scalability up to 256 GPUs
- Dynamic training

Easy deployment

- Java and Scala APIs
- Model Server

Demo:

Image classification with Keras/TensorFlow

+ Script Mode
+ Managed Spot Training
+ Elastic Inference

https://aws.amazon.com/blogs/machine-learning/train-and-deploy-keras-models-with-tensorflow-and-apache-mxnet-on-amazon-sagemaker/

https://gitlab.com/juliensimon/dlnotebooks/tree/master/keras/05-keras-blog-post

Getting started

http://aws.amazon.com/free

https://ml.aws
https://aws.amazon.com/sagemaker

https://github.com/aws/sagemaker-python-sdk
https://github.com/aws/sagemaker-spark
https://github.com/awslabs/amazon-sagemaker-examples

https://gitlab.com/juliensimon/dlnotebooks

Thank you!

Julien Simon
Global Evangelist, AI & Machine Learning
Amazon Web Services
@julsimon

Stéphane Cheikh
Director, Portfolio Evolution using Artificial Intelligence
SITA

aws SUMMIT

Please complete the session survey.

aws SUMMIT