

# Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>



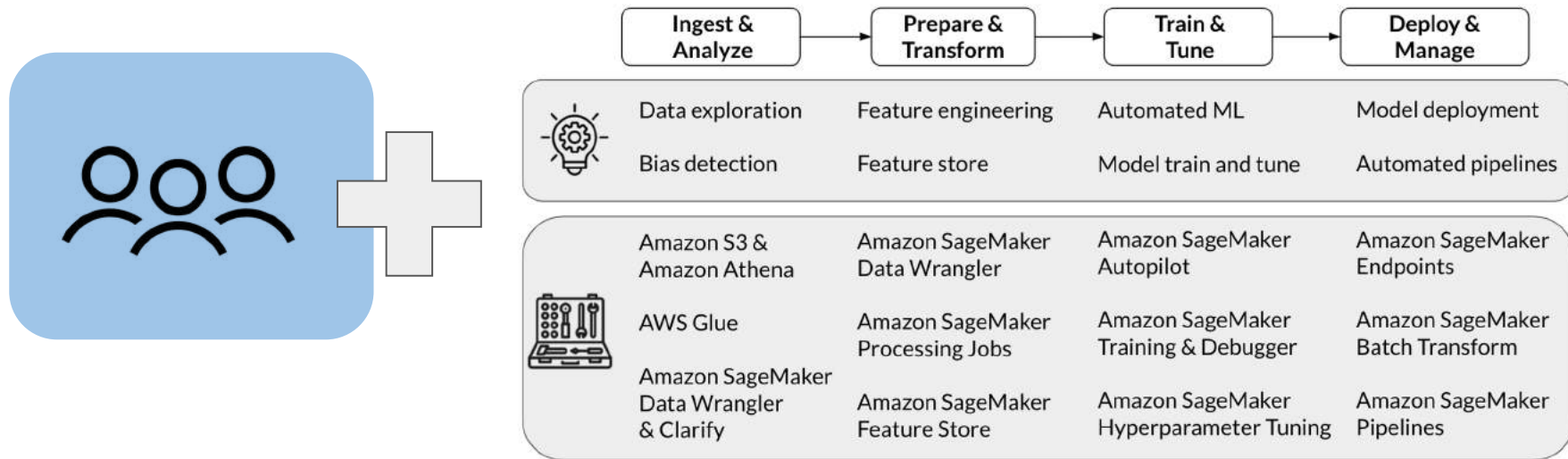
DeepLearning.AI



# Data Labeling and Human-in-the-Loop Pipelines

---

# Human & AI Collaboration



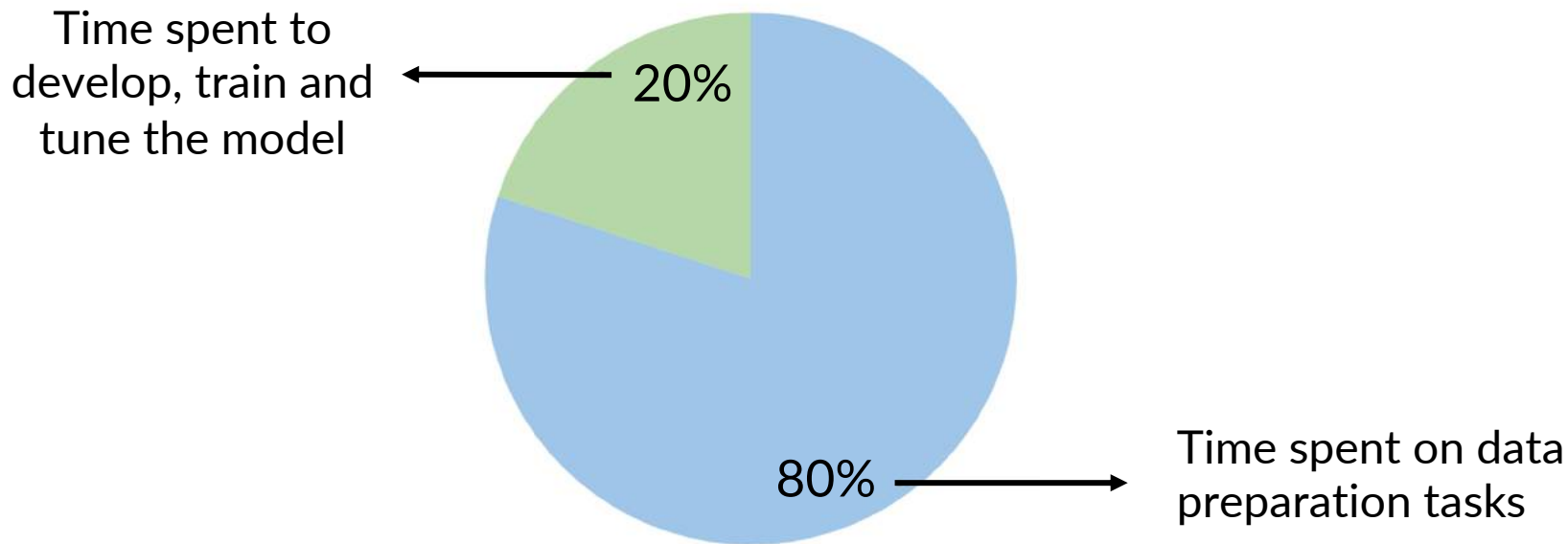
# Human & AI Collaboration



# Data Labeling



# Preparing Data Takes Time



# What Is Data Labeling?

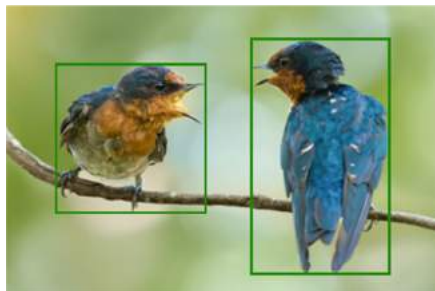
- Process of identifying raw data and adding one or more meaningful and informative labels
- Labels provide context for machine learning models to learn from (supervised learning)
- Correctly labeled datasets are often called "ground truth"



# Common Types Of Data Labeling



# Image Data



Bounding Box



Single-Label Classification



Multi-Label Classification

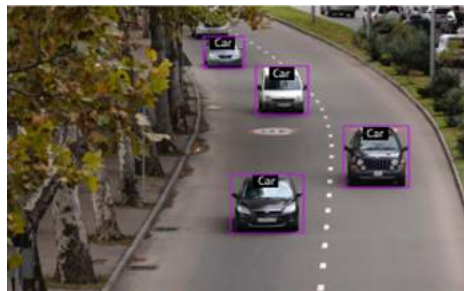


Semantic Segmentation

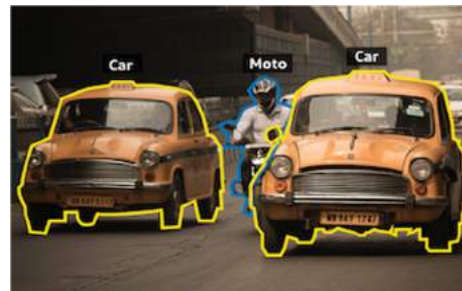
# Video Data



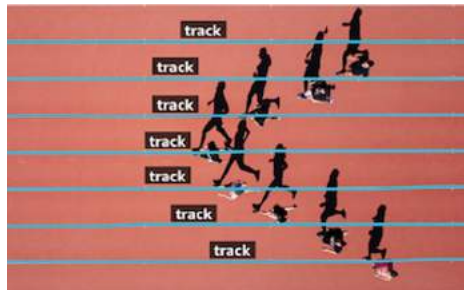
Classification



Bounding box



Polygon



Polyline



Key point

# Text Data



☒ Positive

☐ Negative

*'The movie tells a lovely and wise story with honesty and has been acted out with unassuming grace.'*

Single-Label Classification

☒ Positive

☒ Inspiring

☐ Jargon

*'Every day is a fresh start. Always start your day with a cup of positivitea.'*

Multi-Label Classification

1

2

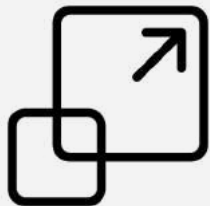
3

4

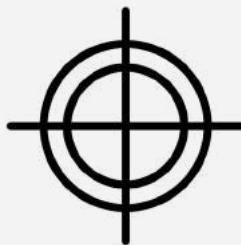
5

Named Entity Recognition

# Challenges



Massive  
scale

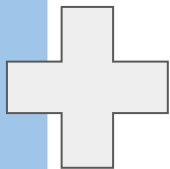
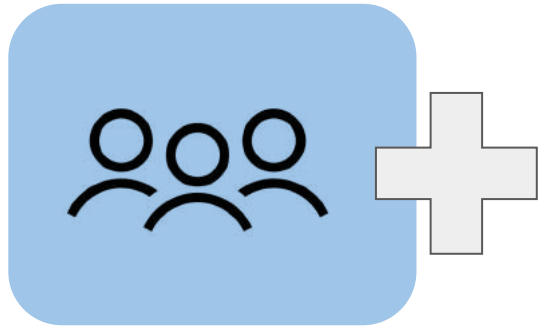


High  
accuracy



Time  
consuming

# How Can Data Labeling Be Done Efficiently?



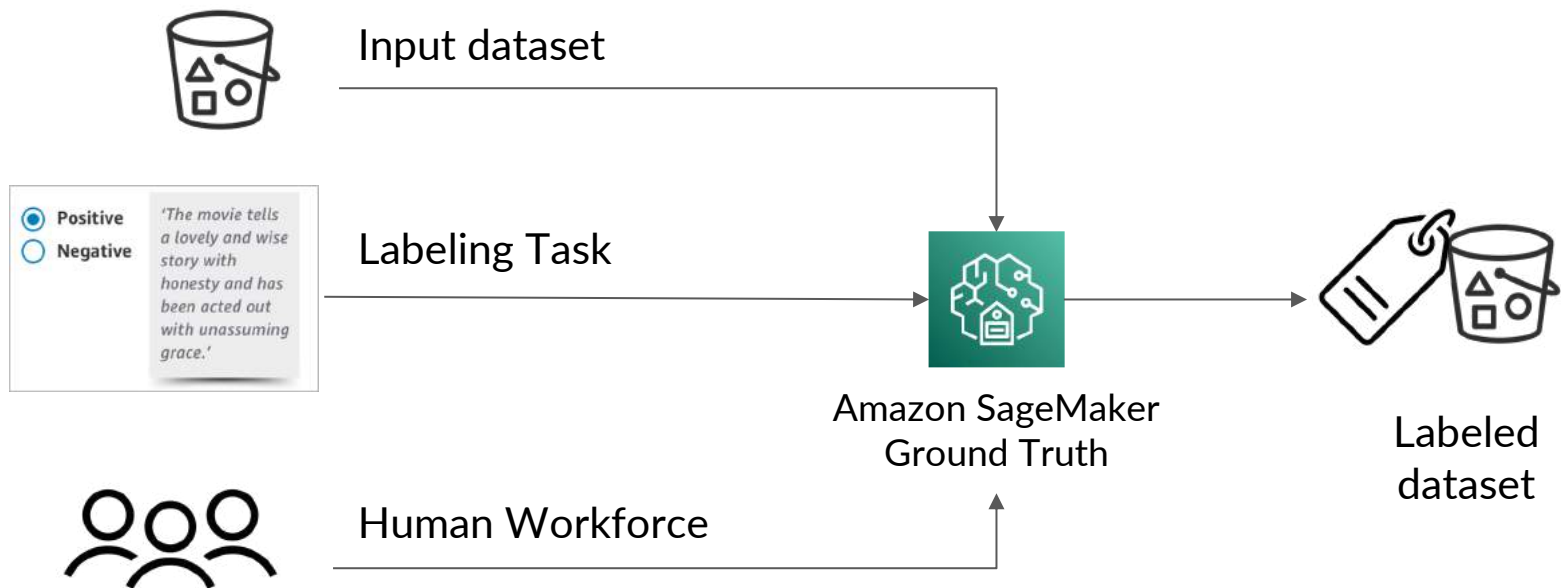
- Access to additional human workforces
- Automated data labeling capabilities
- Assistive labeling features

# Data Labeling

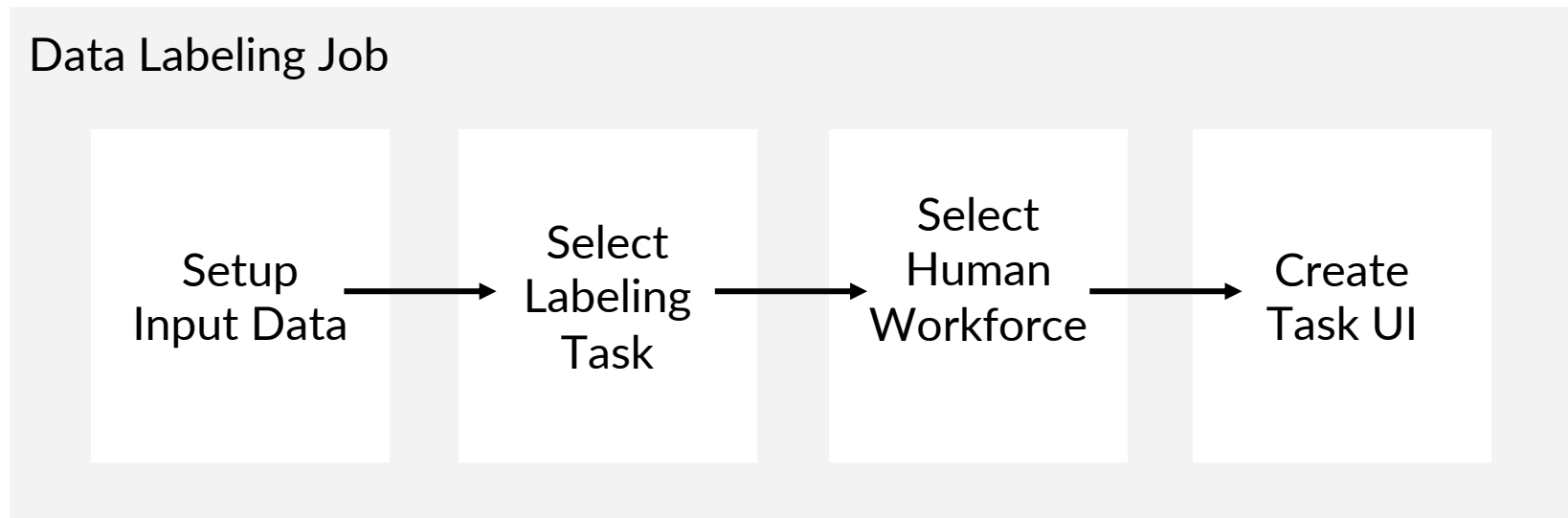
with Amazon SageMaker  
Ground Truth



# Amazon SageMaker Ground Truth

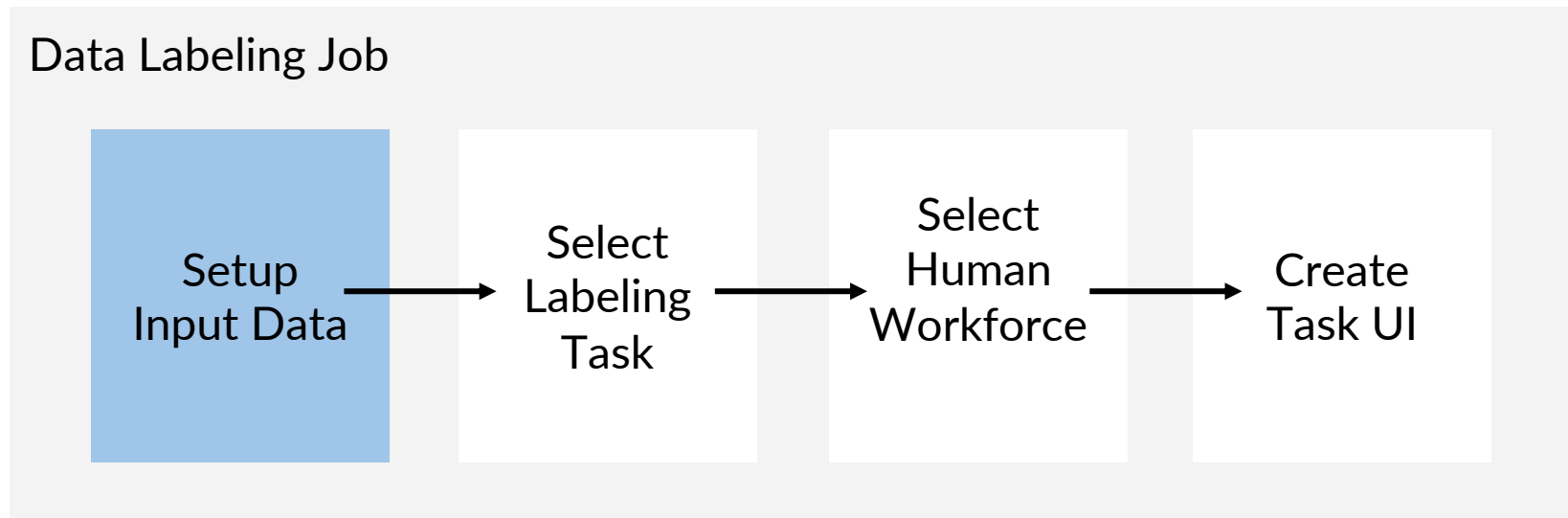


# Steps





# Steps



# Setup Input Data

- Automated data setup
  - Provide the S3 location of the dataset you want labeled

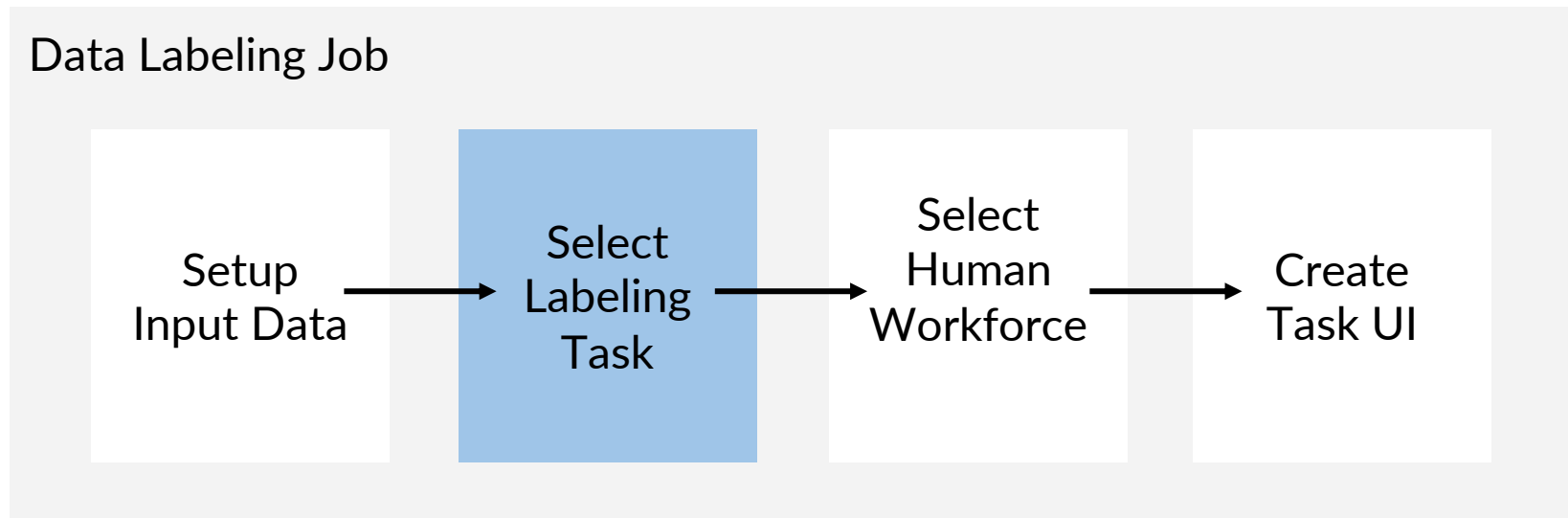
# Setup Input Data

- Automated data setup
  - Provide the S3 location of the dataset you want labeled
- Manual data setup
  - Provide the S3 location of a file (an input manifest file) that identifies the data objects you want labeled

`{"source-ref": "S3 bucket location 1"}`  
...  
`{"source-ref": "S3 bucket location n"}`

`{"source": "I love this product"}`  
...  
`{"source": "It is ok"}`

# Steps



# Select Labeling Task

- Image Data
- Video Data
- Text Data
- Custom

# Select Labeling Task

- Image Data
- Video Data
- Text Data
- Custom

Image Classification (Single Label)

Image Classification (Multi-Label)

Bounding Box

Semantic Segmentation

Label Verification



# Select Labeling Task

- Image Data
- Video Data
- Text Data
- Custom

Video Clip Classification

Video Object Detection/Tracking

Bounding Box

Polygon

Polyline

Key Point



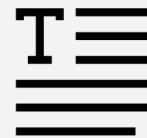
# Select Labeling Task

- Image Data
- Video Data
- Text Data
- Custom

Text Classification (Single Label)

Text Classification (Multi-Label)

Named Entity Recognition

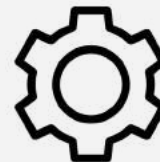




# Select Labeling Task

- Image Data
- Video Data
- Text Data
- Custom

Your custom labeling task



# Custom Labeling Task



AWS Lambda

**Pre-annotation Lambda Function**

Pre-process each data object



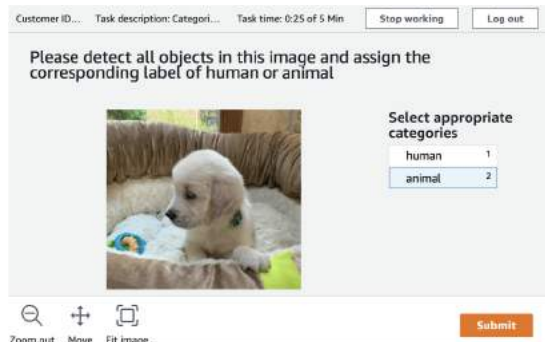
Amazon SageMaker  
Ground Truth



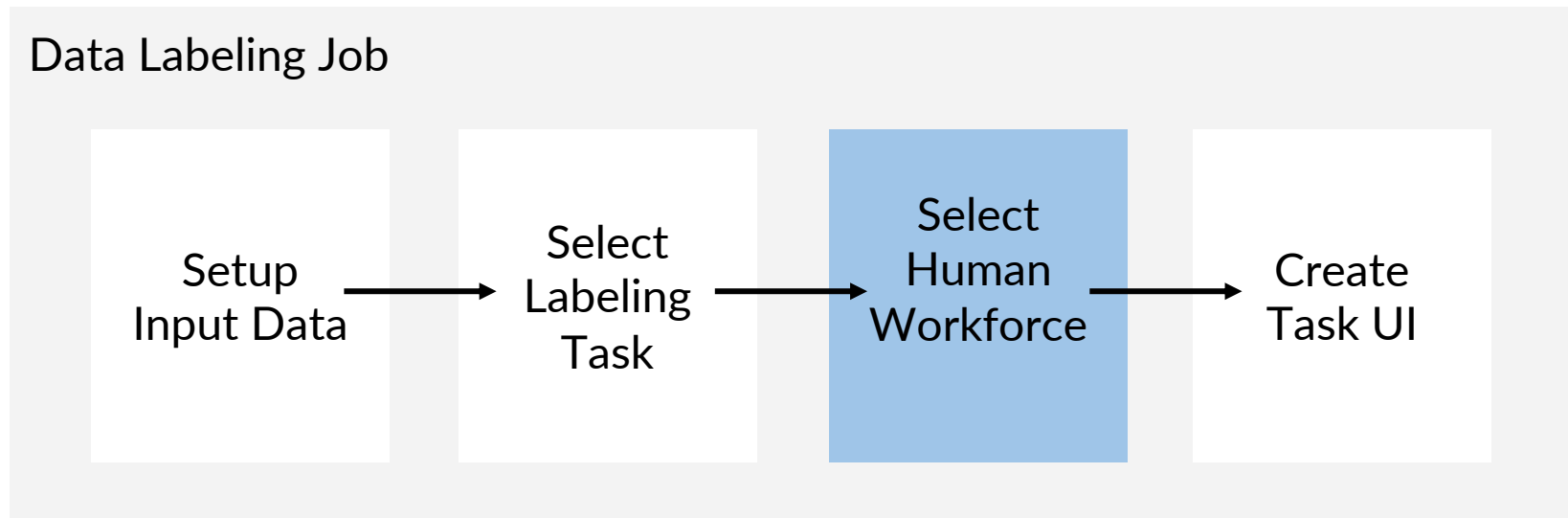
AWS Lambda

**Post-annotation Lambda Function**

Process each labeling result  
Consolidate annotations



# Steps



# Human Workforce Options

Amazon  
Mechanical Turk



Private



You can restrict worker access to tasks to allowable IP addresses

Vendor



# Setup Private Workforce (UI)

The screenshot displays the Amazon SageMaker Studio interface. On the left, a sidebar menu is visible with the following items: Amazon SageMaker Studio, Dashboard, Search, Images, and a collapsed section labeled 'Ground Truth'. The 'Ground Truth' section is expanded, showing 'Labeling jobs', 'Labeling datasets', and 'Labeling workforces'. The 'Labeling workforces' item is highlighted with a blue rounded rectangle. The main content area shows the 'Labeling workforces' page. At the top, there are three tabs: 'Amazon Mechanical Turk', 'Private', and 'Vendor'. The 'Private' tab is selected and highlighted with a blue rounded rectangle. Below the tabs, the 'Private workforce' section is visible, containing the text 'A team of workers from your organization.' and a button labeled 'Create private team'. An arrow points from the 'Private' tab to the 'Create private team' button.

Amazon SageMaker X

Amazon SageMaker Studio

Dashboard

Search

Images

▼ Ground Truth

- Labeling jobs
- Labeling datasets
- Labeling workforces

Amazon SageMaker > Labeling workforces

Amazon Mechanical Turk Private Vendor

Private workforce

A team of workers from your organization.

Create private team

# Setup Private Workforce (UI)

Amazon SageMaker > Labeling workforces > Create private team

## Create private team

### Private team creation

☒ **Create a private team with AWS Cognito**  
Create a private work team by sending email invitations to new workers or importing workers from existing Amazon Cognito user groups.

☐ **Create a private team with OpenID Connect (OIDC)**  
Create a private work team with your own identity provider (IdP). Your IdP must support OIDC user groups.

### Team details

**Team name**  
Give your work team a descriptive name. This name can't be changed later.

Enter a team name

Maximum of 63 alphanumeric characters. Can include hyphens, but not spaces. Must be unique within your account in an AWS Region.

- User authentication via Amazon Cognito or OpenID Connect (OIDC)
- You can connect Amazon Cognito to your enterprise identity provider

# Setup Private Workforce (UI)

**Add workers** [Info](#)

Add workers to your private work team by adding worker email addresses or importing workers from existing Amazon Cognito user groups.

☒ Invite new workers by email ☐ Import workers from existing Amazon Cognito user groups

**Email addresses** [Preview invitation](#)

We send an invitation with instructions to each of the worker email addresses that you add here.

wang@example.com, richard@example.com

Use a comma between addresses. You can add up to 50 workers.

**Organization name**  
We use this information to customize the email that we send to the workers.

Enter the organization name

**Contact email**  
Workers use this address to report issues related to the task.

Enter the contact email

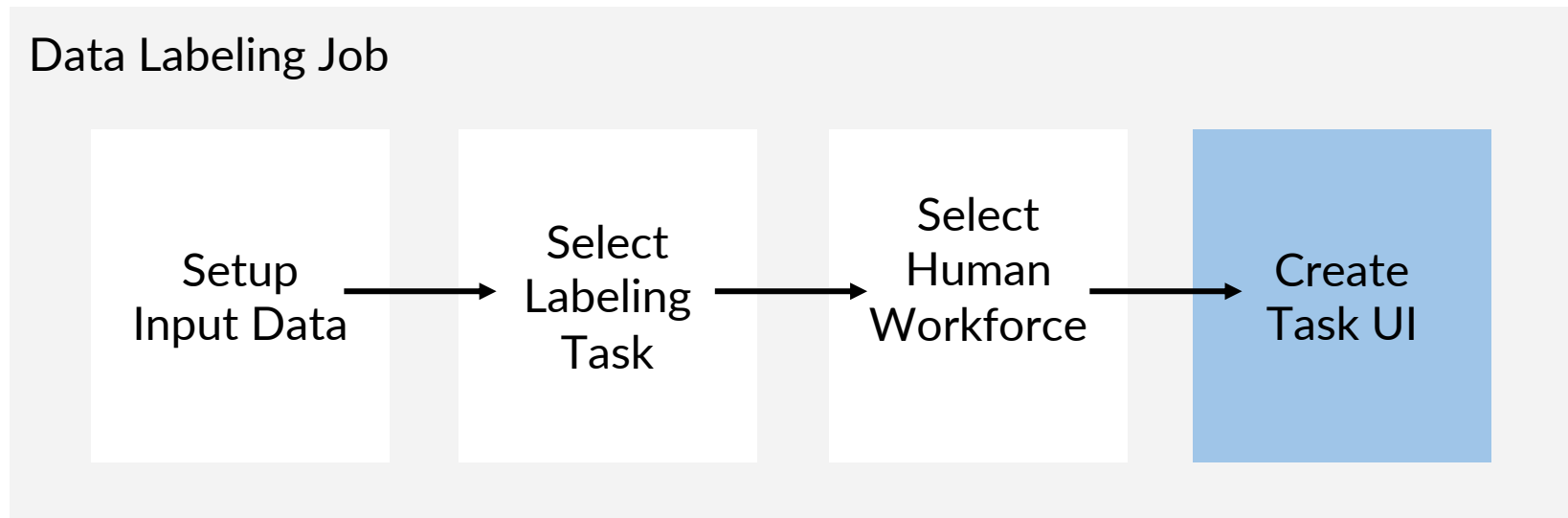
- Invite workers by email
- Enable notifications for work team

**Enable Notifications**

**SNS topic - optional**  
Configuring SNS topic enables your work team to receive notifications on available work. [Learn more](#)

[Cancel](#) [Create private team](#)

# Steps





# Human Task UI

- Combination of HTML, CSS, JavaScript, the Liquid templating language, and Crowd HTML Elements.
- Use built-in templates, or define custom task UI

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>

<crowd-form>
    ...
</crowd-form>
```

# Sample Task UI For Text Classification

```
<script src="https://assets.crowd.aws/crowd-html-elements.js"></script>
<crowd-form>
  <crowd-classifier name="sentiment"
    categories="['-1', '0', '1']"
    initial-value="{{ task.input.initialValue }}"
    header="Classify Reviews into Sentiment: -1 (negative),
                                                    0 (neutral), and
1 (positive)">

    <classification-target>
      {{ task.input.taskObject }}
    </classification-target>

    ...
```

Customize instructions  
to match product reviews  
text classification

# Sample Task UI For Text Classification

```
...
  <full-instructions header="Classify reviews into sentiment:
                                -1 (negative), 0
(neutral), and 1 (positive)">
    <p><strong>1</strong>: joy, excitement, delight</p>
    <p><strong>0</strong>: neither positive or negative</p>
    <p><strong>-1</strong>: anger, sarcasm, anxiety</p>
  </full-instructions>
  <short-instructions>
    Classify reviews into sentiment:
      -1 (negative), 0 (neutral), and 1 (positive)
  </short-instructions>
</crowd-classifier></crowd-form>
```

Provide more detailed instructions.

# Define Task UI For Text Classification

Hello, user-

Cust...

Task description:...

Task time: 2:34 of 60 Min

Decline task

Release task

Stop and resume later

Instructions

Shortcuts

Classify Reviews into Sentiment: -1 (negative), 0 (neutral), and 1 (positive)

Instructions

×

Classify reviews into sentiment: -1 (negative), 0 (neutral), and 1 (positive)

I am unhappy with this product

Select an option

-1	1
0	2
1	3

Human labeler follows instructions to label data presented here.

Submit

Treat the data in this task as confidential.

# Data Labeling

Best Practices



# Best Practices

- Provide clear instructions
- Consolidate annotations to improve label quality
- Verify and adjust labels
- Use automated data labeling on large datasets (Active Learning)
- Re-use prior labeling jobs to create hierarchical labels


# Provide Clear Instructions

### Instructions

[View full instructions](#)


[View tool guide](#)

**Good example**



Each box should be as small as possible.

**Bad example**




Each flower should have one box.


### Bounding box instructions

If the flower touches the edge of the picture, draw the box to the edge of the picture.

**Good Example**




**Bad Example**




For overlapping flowers, draw one box for each. Don't include parts of the flower you can't see.

**Good Example**



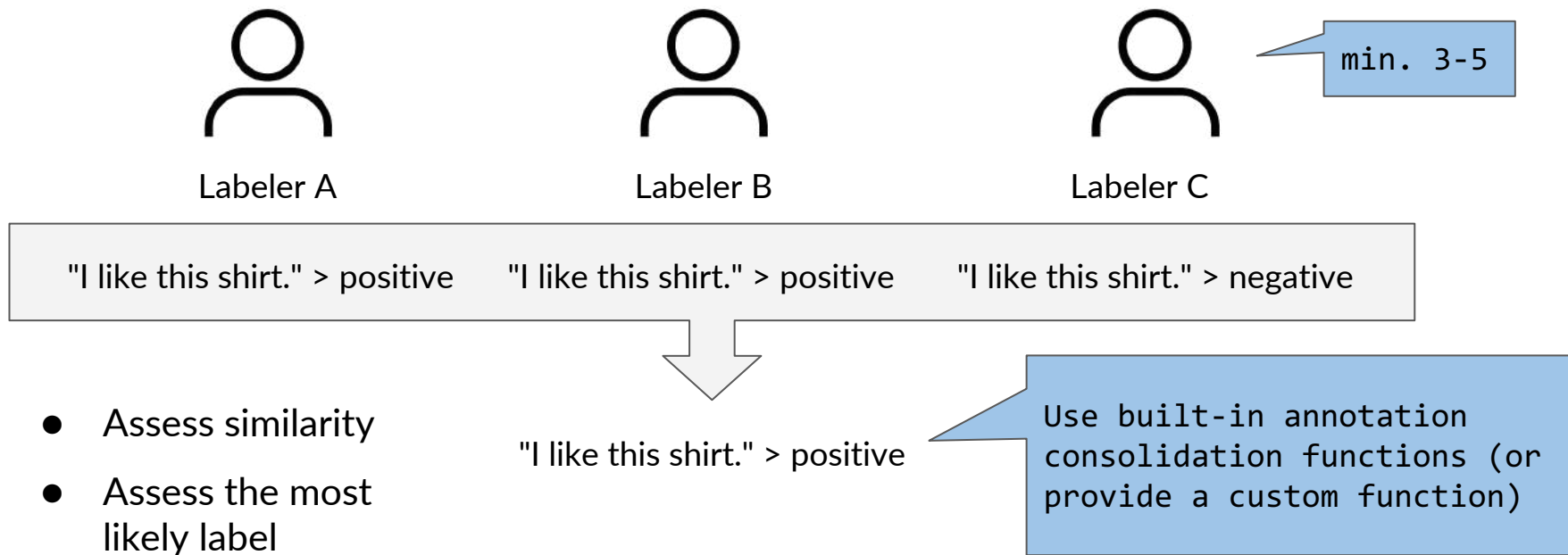
**Bad Example**



Close

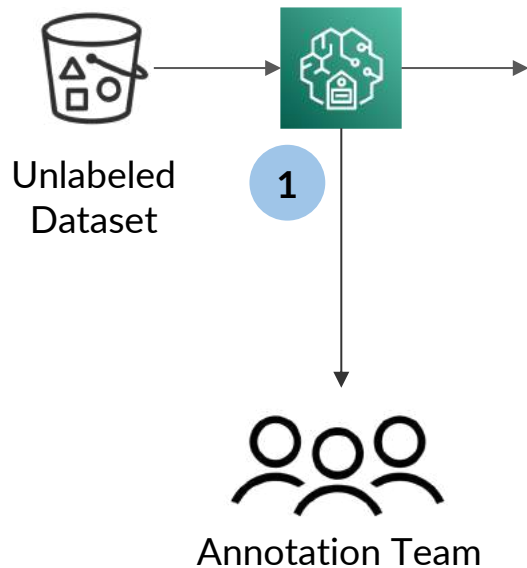
- Provide examples of good and bad annotations
- Only show relevant labels
- Use multiple workers per task

# Consolidate Annotations To Improve Label Quality

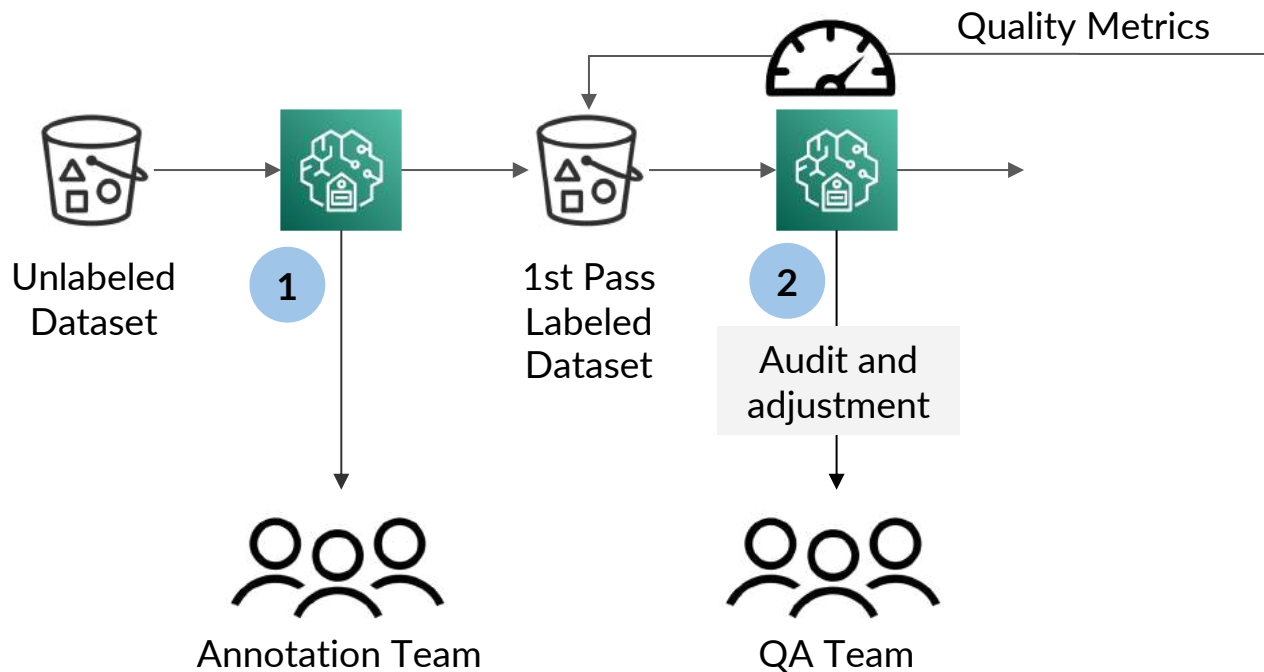




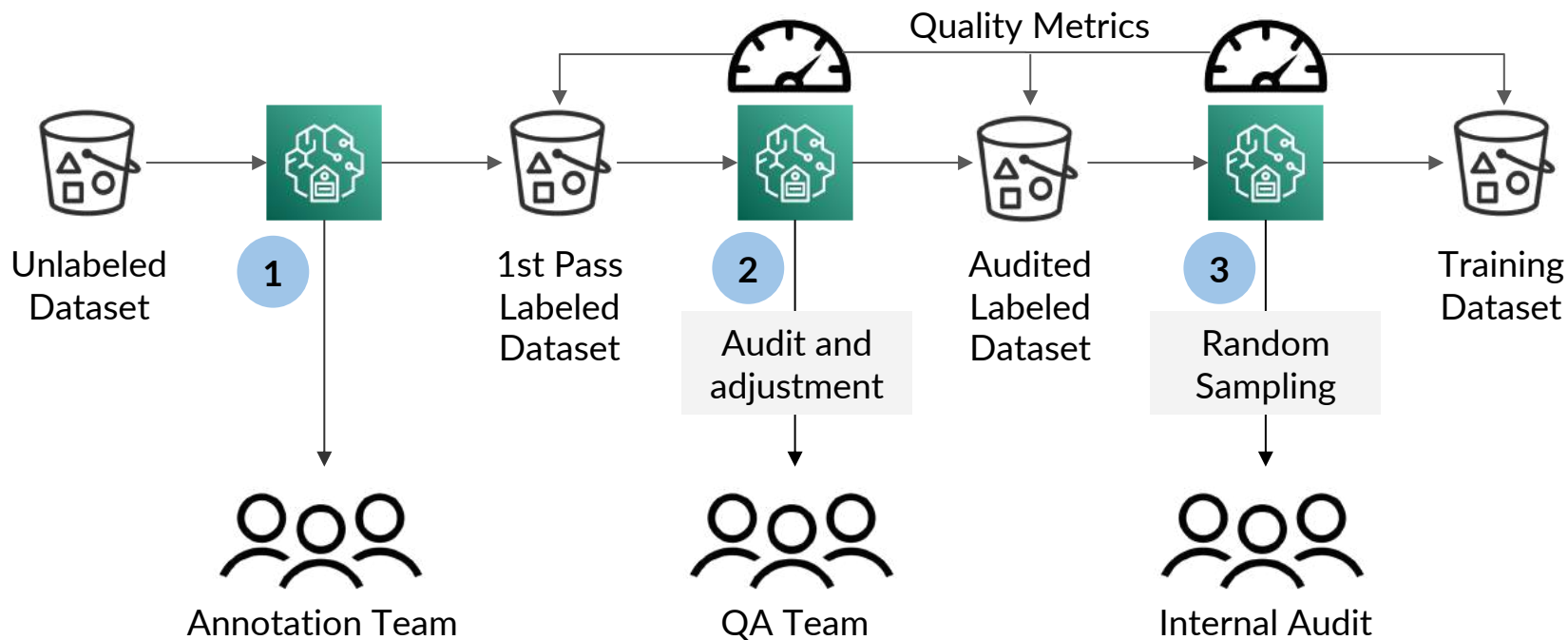
# Verify And Adjust Labels



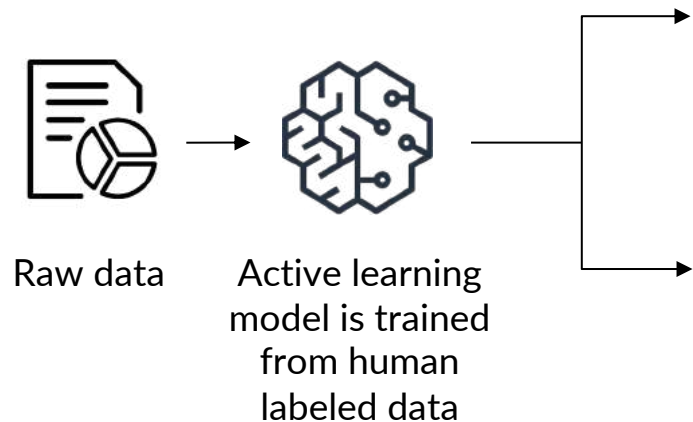
# Verify And Adjust Labels



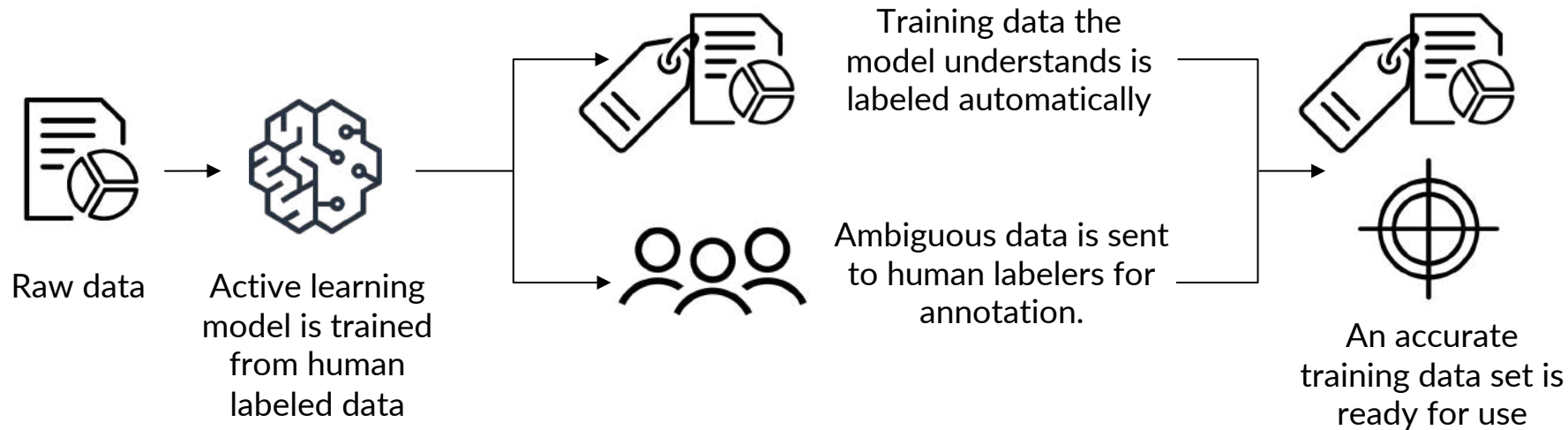
# Verify And Adjust Labels



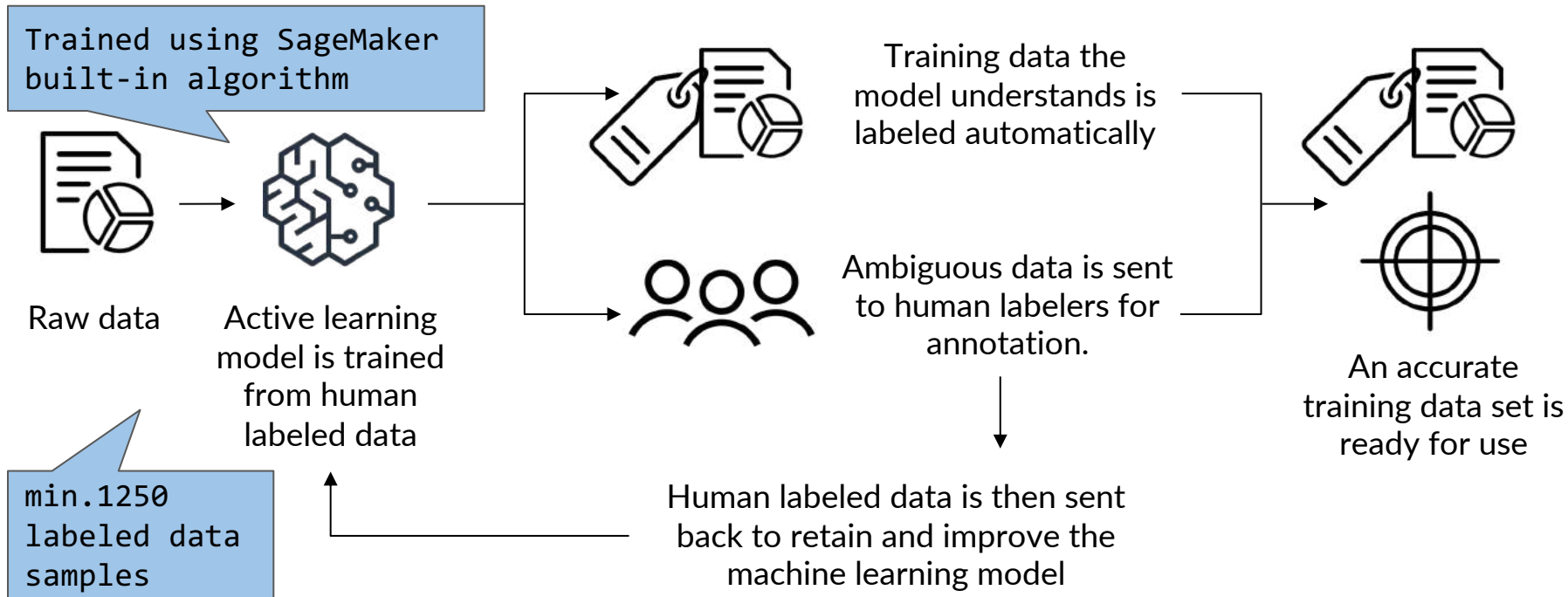
# Use Automated Data Labeling On Large Datasets



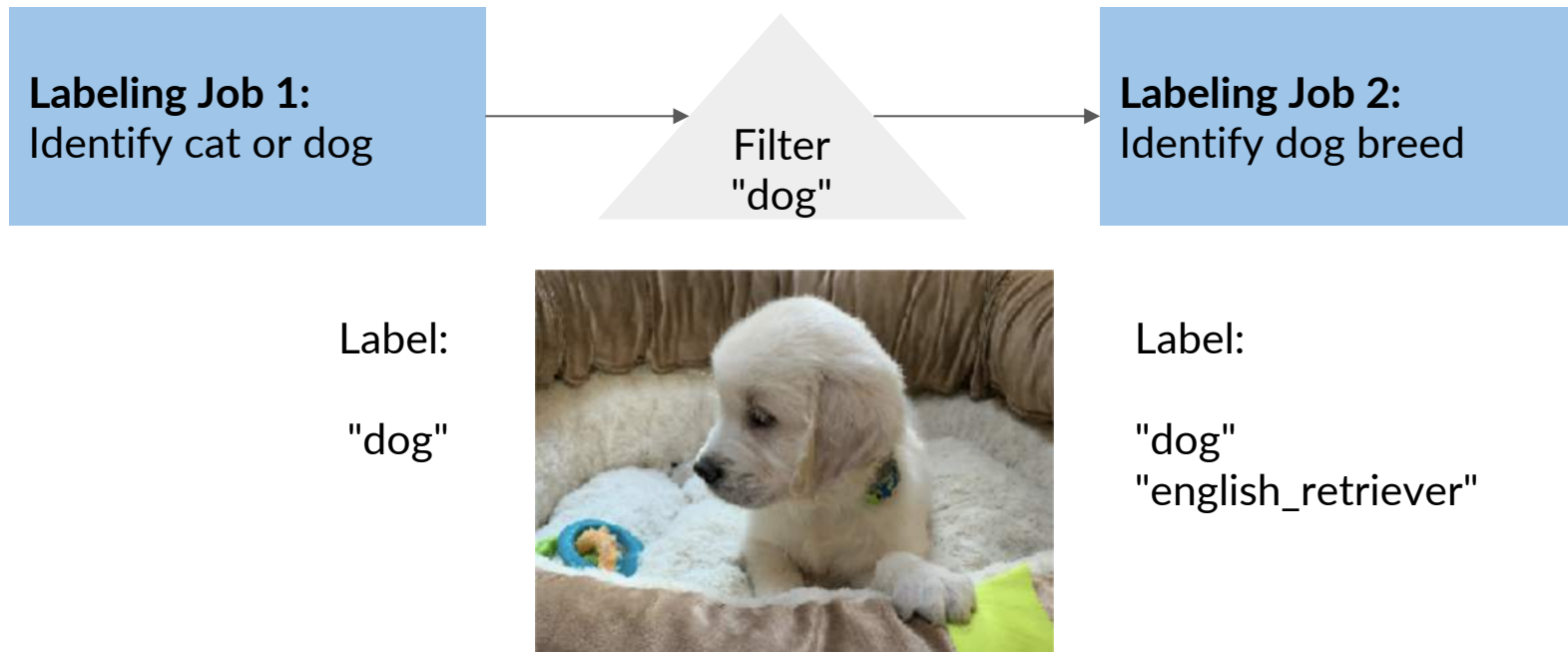
# Use Automated Data Labeling On Large Datasets



# Use Automated Data Labeling On Large Datasets



# Re-Use Prior Labeling Jobs To Create Hierarchical Labels



# Human-In-The-Loop Pipelines





# Image Classification

**Task:** *Verify the appropriate dog breed has been classified in the image*

1. Read the task carefully and inspect the image.
2. Read the options and review the examples to understand more about the dog breed categories.
3. Choose the appropriate dog breed category that best suits the image.

## Examples

Corgi



Chihuahua



# Form Extraction

**Task: Review the key-value pairs to the right and correct them if they do not match the document**

[View full instructions](#)  
[View tool guide](#)

Click on a key-value block or input box to highlight the corresponding key-value pair in the document.

If it is a key-value pair, review the content for the key or value. If the content is incorrect, correct it.

If it's not a key-value relationship, mark it **No**.

Jane Doe

123 Any Street,  
Any Town,  
USA

Key-value pair  
☐ Yes ☒ No  
Jane Doe  
123 Any Street,  
☐ Value is blank  
☐ Key is not found

If you can't find the key in the document, mark **Key is not found**.

Mail address

☒ Key is not found  
☐ Value is blank

If the content of a field is empty, mark **Value is blank**.

Cell number

## Employment Application

### Application Information

Full Name

Jane Doe

Phone number:

550-0100

Home address:

123 Any Street, Any Town, USA

Mail address:


same as home address


Key-value pair  
☒ Yes ☐ No  
Full name:  
Jane Doe  
☐ Value is blank  
☐ Key is not found


Key-value pair  
☒ Yes ☐ No  
Phone number:  
550-0100  
☐ Value is blank  
☐ Key is not found


☐ No adjustment needed

Submit

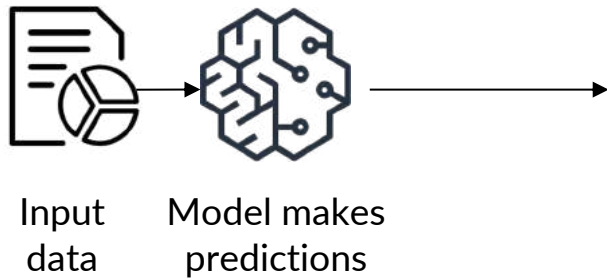
 Zoom in

 Zoom out

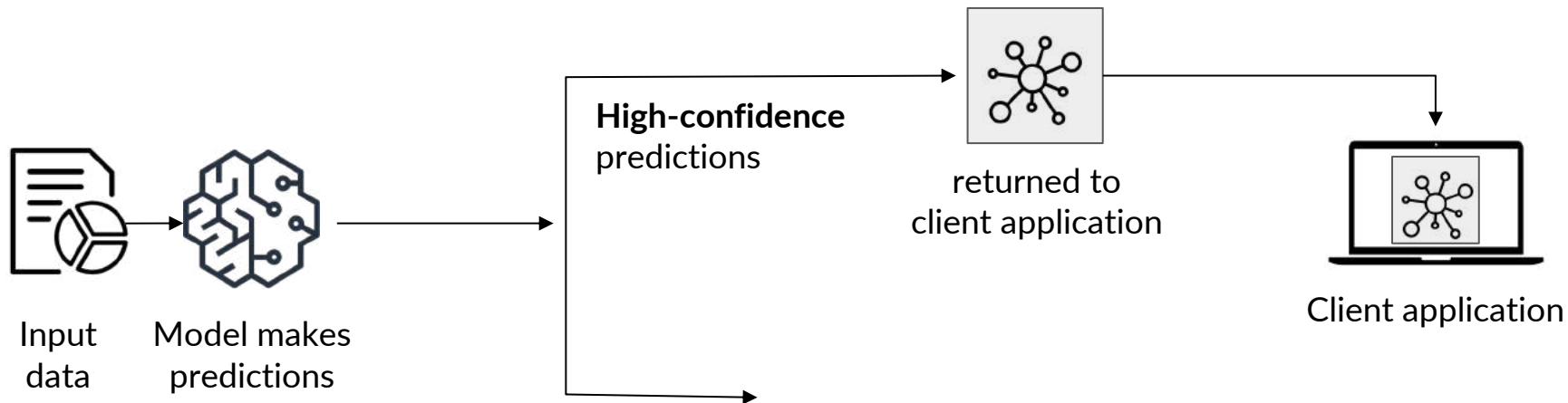
 Move

 Fit image

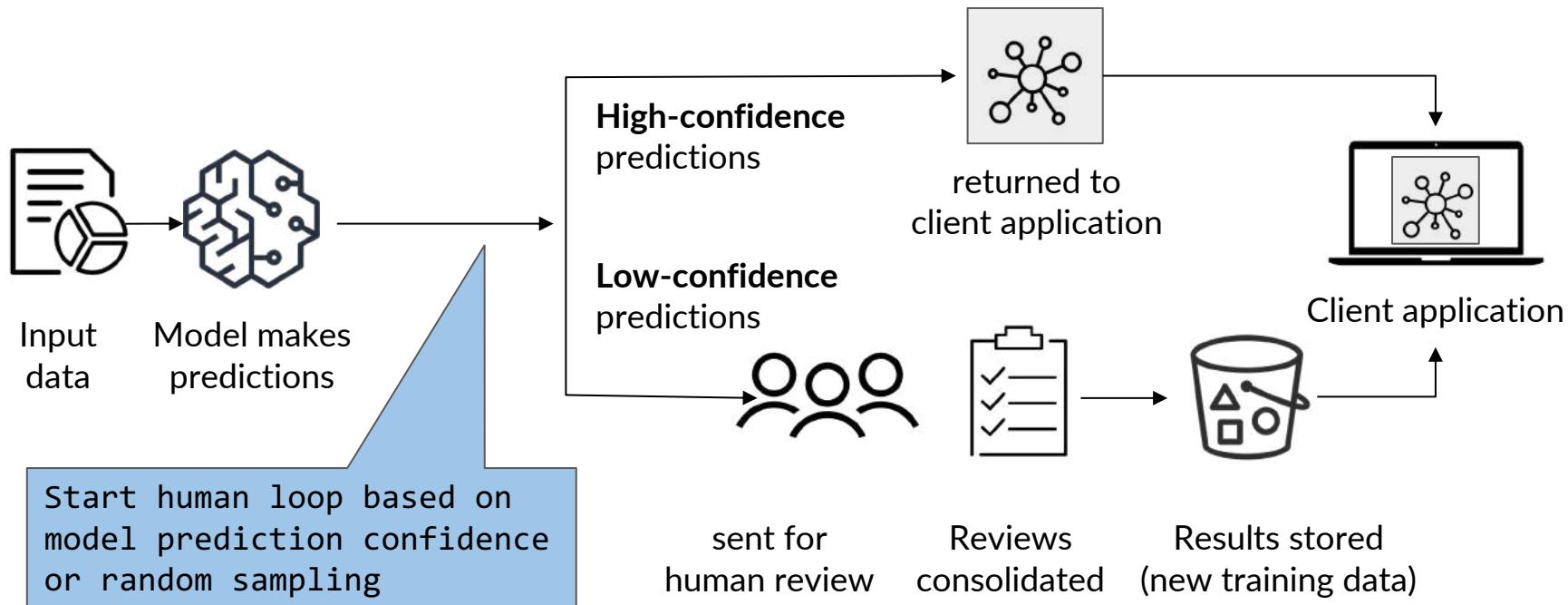
# Implement Human Review Of Model Predictions



# Implement Human Review Of Model Predictions



# Implement Human Review Of Model Predictions



# Challenges

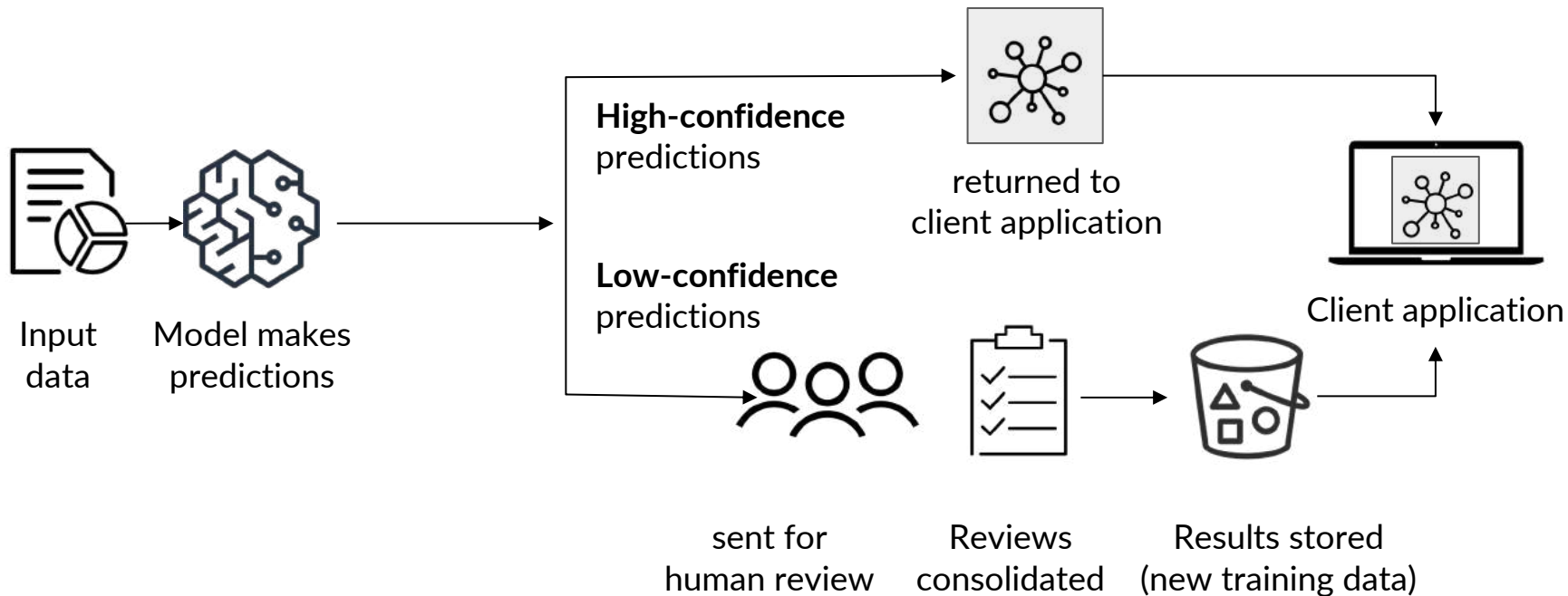
- Need ML scientists, engineering, and operations teams
- Need to manage large number of reviewers
- Need to write custom software to manage review tasks
- Difficult to achieve high review accuracy

# Human-In-The-Loop Pipelines

with Amazon Augmented AI  
(Amazon A2I)

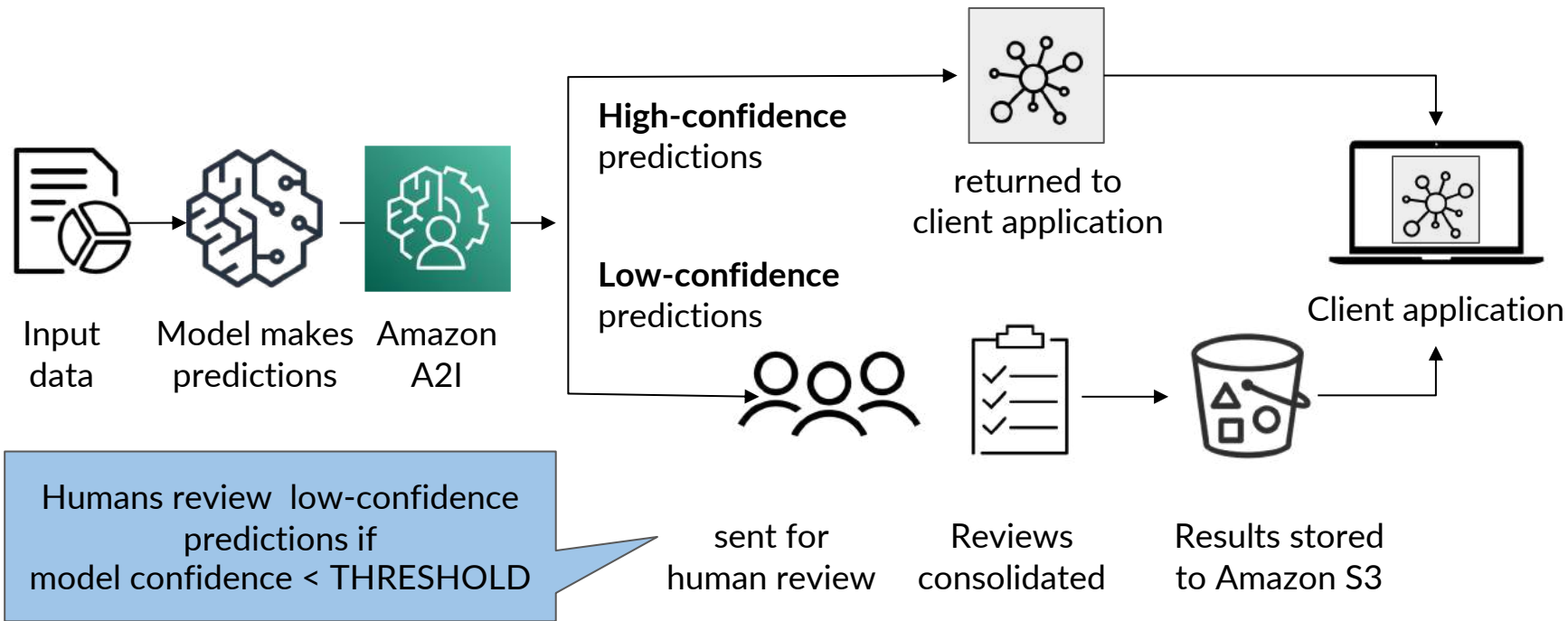


# Implement Human Review Of Model Predictions

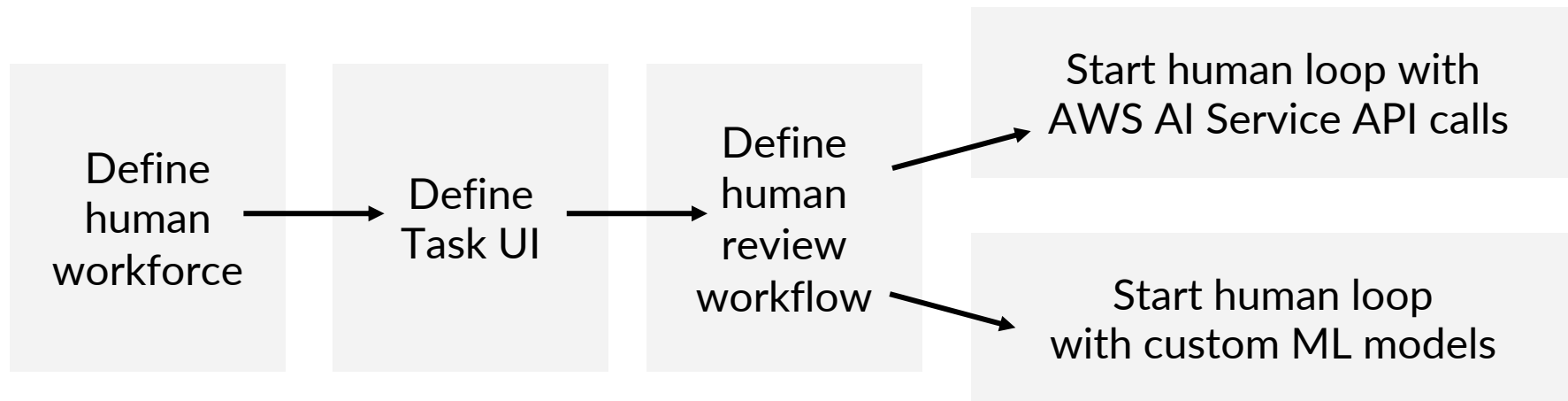




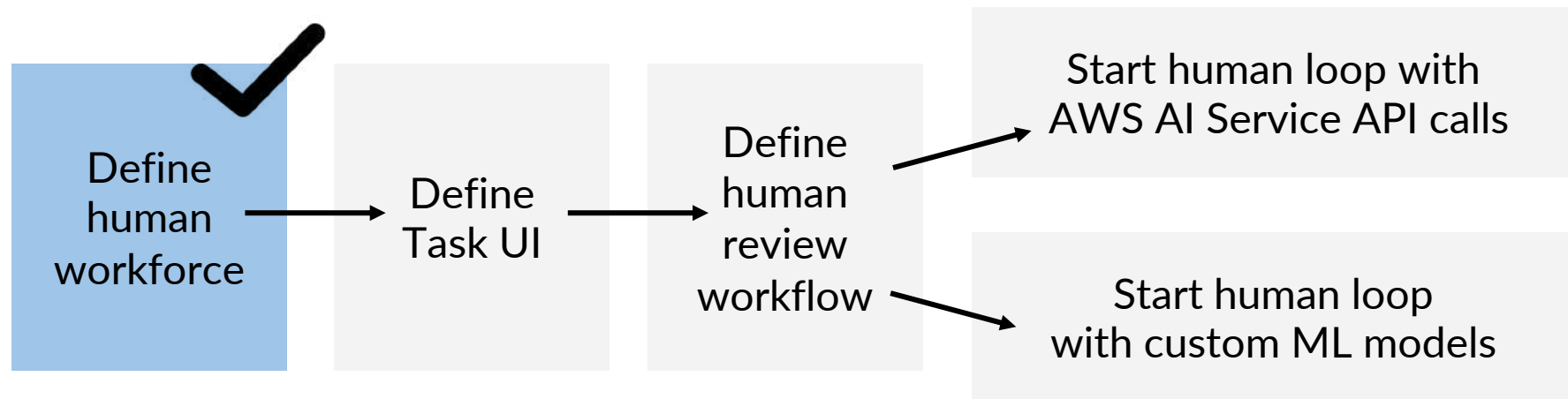
# Amazon A2I



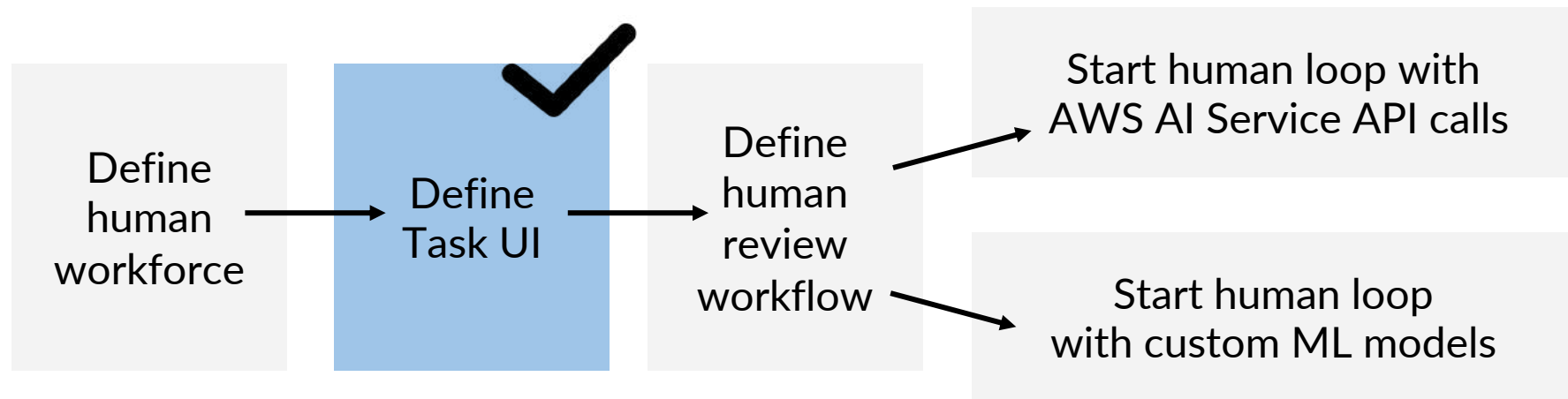
# Steps



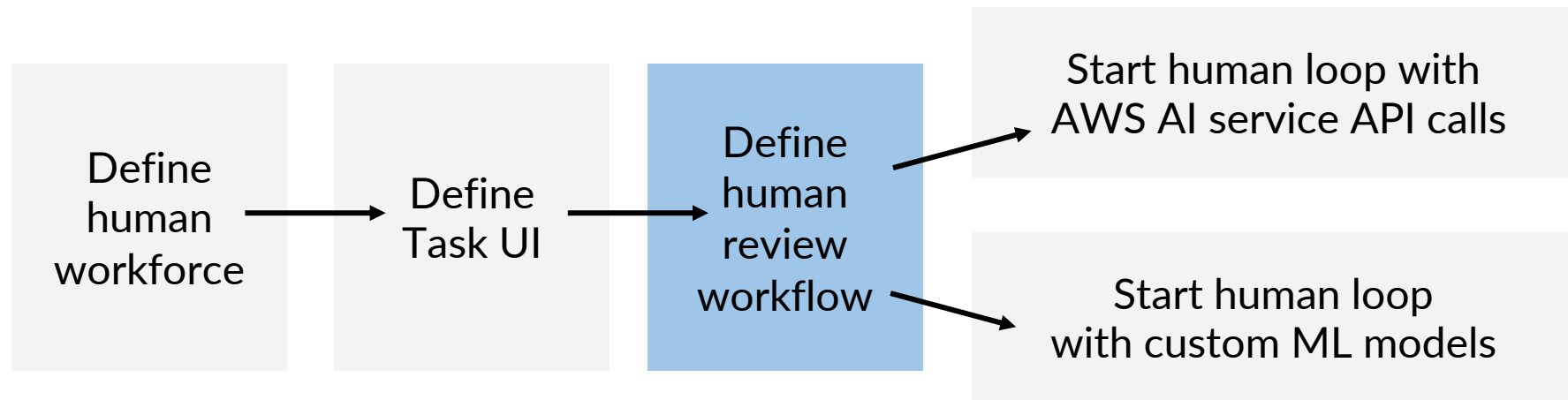
# Steps



# Steps



# Steps



# Define Human Review Workflow

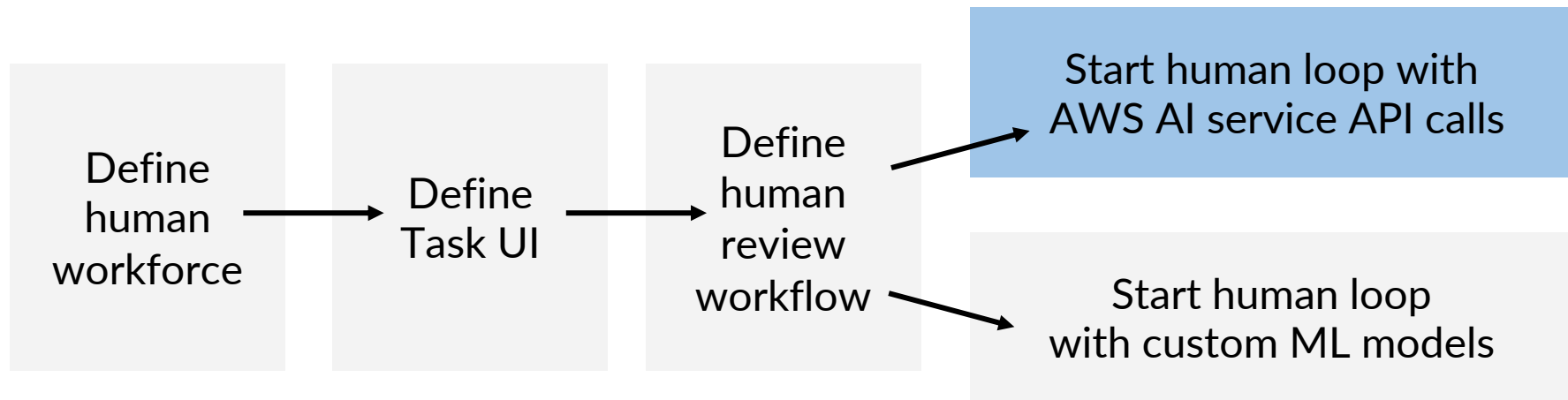
```
create_workflow_definition_response = sm.create_flow_definition(  
    FlowDefinitionName=<NAME>,  
    RoleArn=role,  
    HumanLoopConfig={  
        "WorkteamArn": ...,  
        "HumanTaskUiArn": ...,  
        "TaskCount": 1,  
        "TaskDescription": "Classify Reviews into sentiment: -1 (negative),  
                           0 (neutral), 1  
(positive)",  
        "TaskTitle": ... },  
    OutputConfig={"S3OutputPath": output_path}, ... )  
  
augmented_ai_flow_definition_arn =  
    create_workflow_definition_response["FlowDefinitionArn"]
```

Workforce and team to use

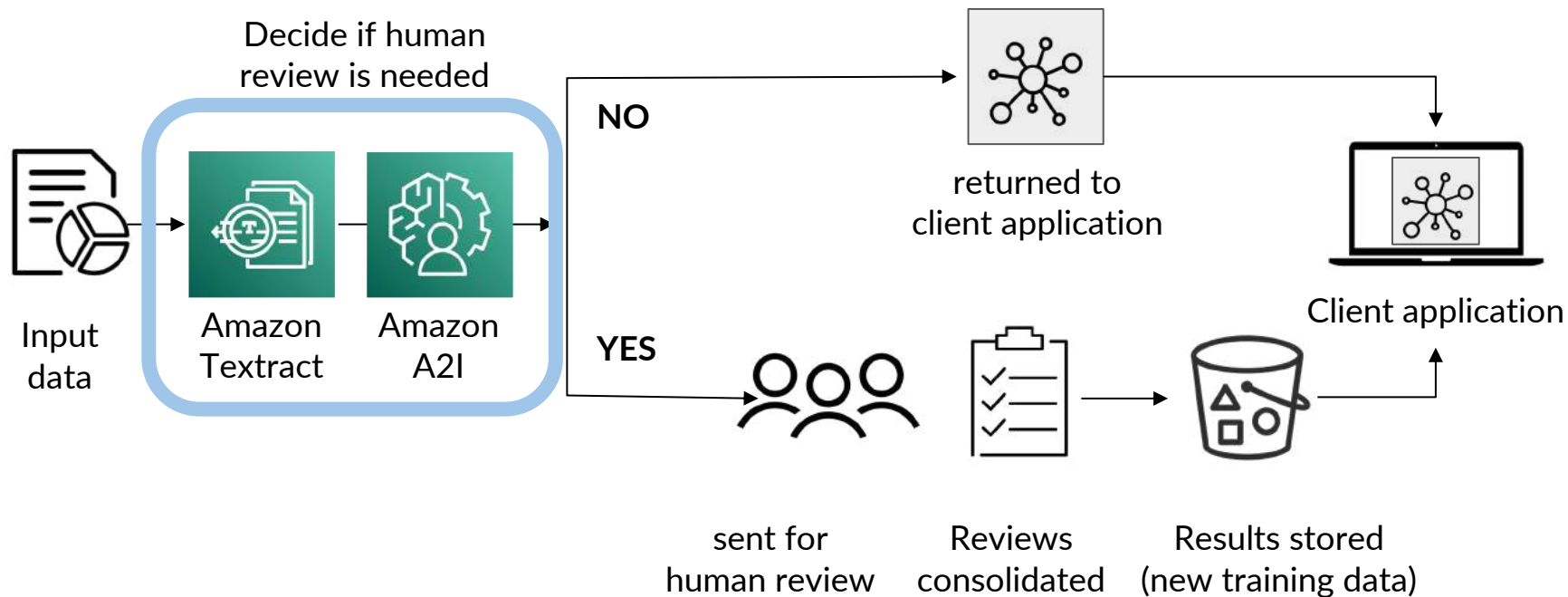
Human Task UI and additional configuration

Where to store output data

# Steps



# Start Human Loop With Amazon Textract





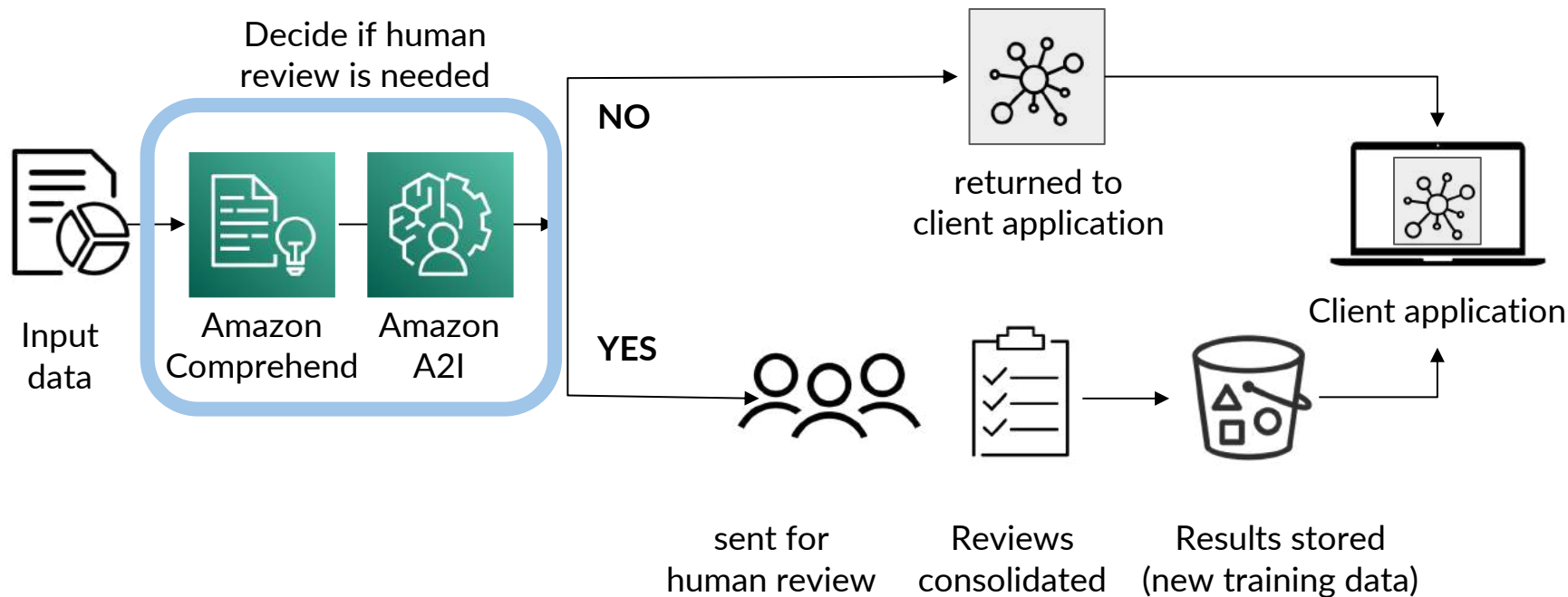
# Start Human Loop With Amazon Textract

```
def analyze_document_with_a2i(document_name, bucket):  
    response = textract.analyze_document(  
        Document={'S3Object': {'Bucket': bucket, 'Name': document_name}},  
        ...  
        HumanLoopConfig=humanLoopConfig  
    )  
    return response
```

Add your human loop configuration to the Amazon Textract API call

```
analyzeDocumentResponse = analyze_document_with_a2i(document, S3_BUCKET)
```

# Start Human Loop With Amazon Comprehend



# Start Human Loop With Amazon Comprehend

```
reviews = ["I enjoy this product", "It is okay"]  
CONFIDENCE_SCORE_THRESHOLD = 0.90
```

Sample reviews and  
confidence score threshold

```
for review in reviews:
```

```
    # Call the a Amazon Comprehend Custom model
```

```
    response = comprehend.classify_document(  
        Text=review,  
        EndpointArn=comprehend_endpoint_arn)
```

Get prediction response  
from a custom Amazon  
Comprehend model

```
    sentiment = response["Classes"][0]["Name"]  
    confidence_score = response["Classes"][0]["Score"]  
    print(f'Processing sample_review: "{review}")  
    ...
```

Parse sentiment class and  
confidence score from  
model response

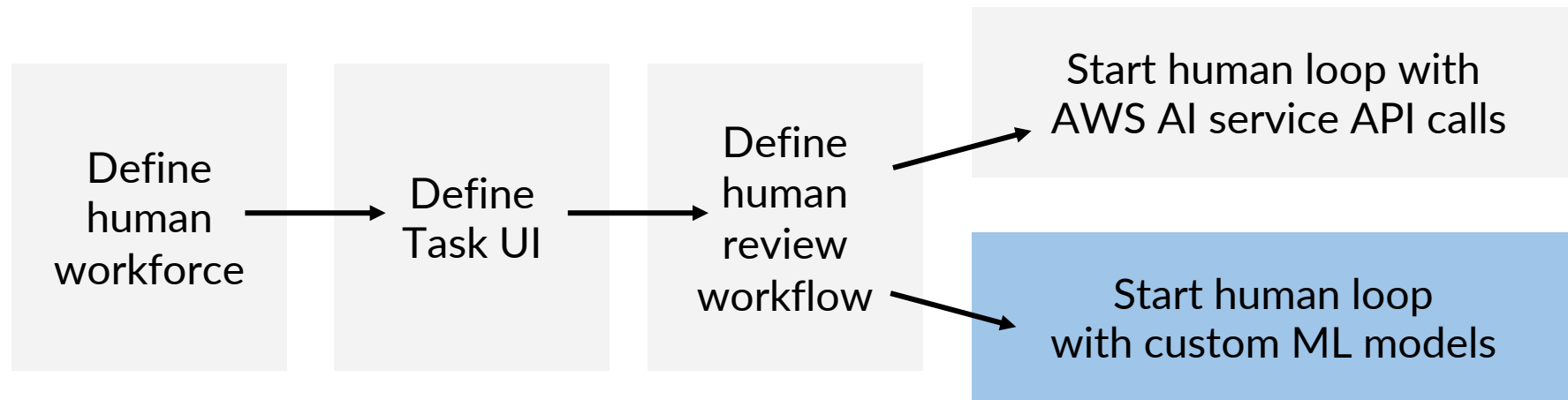
# Start Human Loop With Amazon Comprehend

```
...  
# Condition when to engage a human for review  
if confidence_score < CONFIDENCE_SCORE_THRESHOLD:  
    humanLoopName = <NAME>  
    inputContent = {"initialValue": sentiment, "taskObject": sample_review}  
  
start_loop_response = a2i.start_human_loop(  
    HumanLoopName=humanLoopName,  
    FlowDefinitionArn=augmented_ai_flow_definition_arn,  
    HumanLoopInput={"InputContent": json.dumps(inputContent)},  
)
```

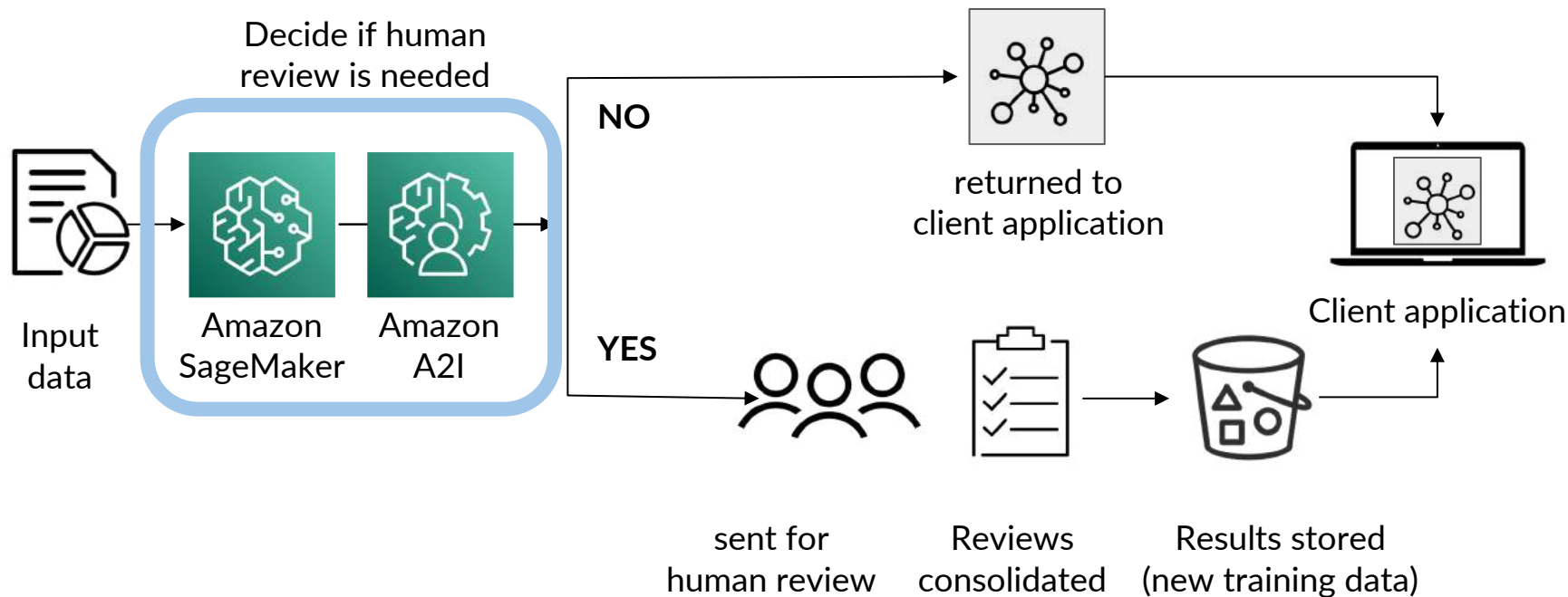
Check returned  
confidence score

Start Amazon A2I human  
loop with inputs

# Steps



# Use Amazon A2I With Amazon SageMaker



# Use Amazon A2I With Amazon SageMaker

```
from sagemaker.predictor import Predictor
from sagemaker.serializers import JSONLinesSerializer
from sagemaker.deserializers import JSONLinesDeserializer
```

```
class SentimentPredictor(Predictor):
    def __init__(self, endpoint_name, sagemaker_session):
        super().__init__(endpoint_name,
                          sagemaker_session=...,
                          serializer=...,
                          deserializer=...)
```

Specify how to process model inputs (serializer) and model outputs (deserializer).

# Use Amazon A2I With Amazon SageMaker

```
from sagemaker.pytorch.model import PyTorchModel  
pytorch_model_name = 'model-{}'.format(timestamp)
```

Specify custom  
predictor

```
model = PyTorchModel(name=...,
```

```
predictor_cls=...,
```

```
model_data='s3://.../model.tar.gz',  
...)
```

S3 location of  
model artifact

```
pytorch_endpoint_name = 'endpoint-{}'.format(timestamp)
```

Deploy model endpoint

```
predictor = model.deploy(endpoint_name=pytorch_endpoint_name, ...)
```



# Use Amazon A2I With Amazon SageMaker

```
reviews = ["I enjoy this product", "It is okay"]
```

Sample reviews

```
for review in reviews:
```

```
    inputs = [  
        {"features": [review]},  
    ]
```

Call the model through  
the predictor.predict()

```
    response = predictor.predict(inputs)
```

```
    prediction = response[0]['predicted_label']  
    confidence_score = response[0]['probability']  
    ...
```

Parse predicted label and  
confidence score from  
model response

# Use Amazon A2I With Amazon SageMaker

```
...  
if confidence_score < CONFIDENCE_SCORE_THRESHOLD:  
  
    human_loop_name = <NAME>  
    input_content = {"initialValue": ..., "taskObject": ...}  
  
    start_loop_response = a2i.start_human_loop(  
        HumanLoopName=human_loop_name,  
        FlowDefinitionArn=augmented_ai_flow_definition_arn,  
        HumanLoopInput={"InputContent": json.dumps(input_content)},  
    )  
...
```

Check returned  
confidence score

Start Amazon A2I  
human loop with  
inputs

# Human Loops Started

```
[{'probability': 0.937, 'predicted_label': 1}]
```

Checking prediction confidence 0.937 for sample review: "I enjoy this product"

Confidence score of 93.7% for star rating of 1 is above threshold of 90.0%

No human loop created.

```
[{'probability': 0.542, 'predicted_label': 1}]
```

Checking prediction confidence 0.542 for sample review: "It is okay"

Confidence score of 54.2% for prediction of 1 is less than the threshold of 90.0%

\*\*\* ==> Starting human loop with name: 1620962189-045527

# Verify Results

input_content	human_answer
<code>{'initialValue': 1, 'taskObject': 'It is okay'}</code>	<code>{'sentiment': {'label': '0'}}</code>