



AWS  
re:Invent

**API305-R**

# Building serverless machine-learning workflows

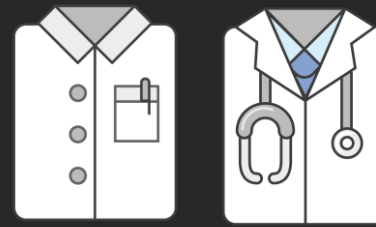
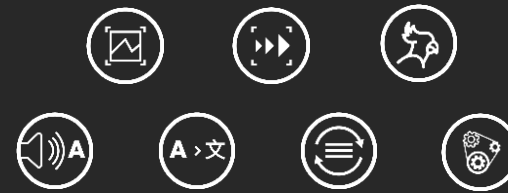
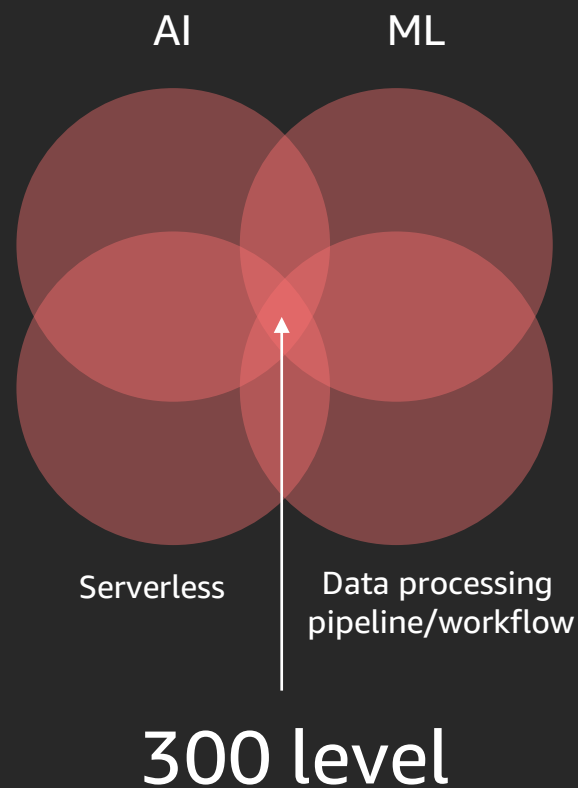
**Mark Roy**

Machine Learning Specialist SA  
Amazon Web Services

**Ibrahim Gabr**

Machine Learning Specialist SA  
Amazon Web Services

# Who is this chalk talk for?



## AI developers

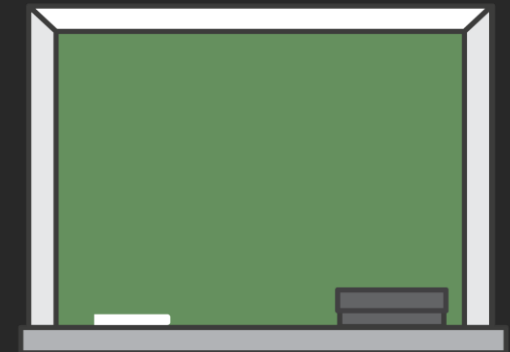
You use Amazon Rekognition, Amazon Comprehend

## Data engineers

You build data processing pipelines or automate data-related tasks

## ML scientists

You build your own models using Amazon SageMaker



## Chalk talk

Ask questions!

# Agenda

- Data and machine learning
- Monolith versus serverless
- Orchestration patterns
- Demo

# Related sessions

**SVS214-R** – Amazon S3 to Lambda: A flexible pattern at the core of serverless applications

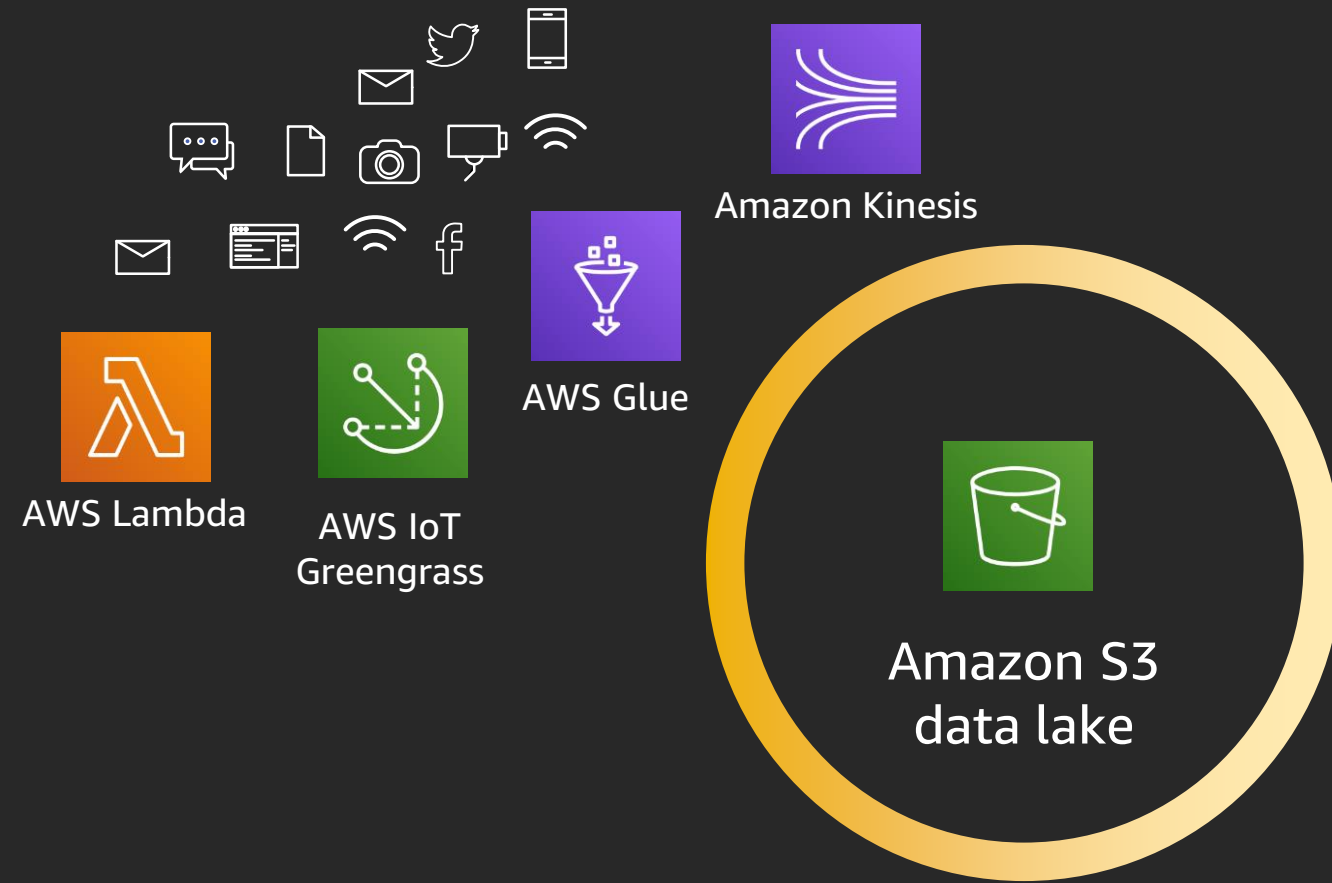
**AIM206-R** – Finding a needle in a haystack: Use AI to transform content management

**AIM211-R** – AI document processing for business automation

**AIM305-R** – Automate content moderation and compliance with AI

**AIM340-R** – Next-gen video: Transcription, translation & search, powered by ML

# Streams to lakes



# Using the right tools for analysis



# Preparing output for use





# Leveraging AI services



# Monolith vs. Step Functions

	Monolith	Step Functions
Error handling	Complex retry logic	Built-in constructs for exponential back-off
Parallel tasks	Hard-coded, runtime-specific	Native states for parallel tasks
Orchestration	Business logic intertwined, difficult to visualize	Separated orchestration logic, easier to maintain and debug, visual state machine
Extensibility	Surgey to large Lambda functions	Easier addition of new states


# Common orchestration patterns

1. Orchestration of multiple services
2. Asynchronous processes
3. Child workflows
4. Data available at all stages

# Common pattern #1: Orchestration of multiple services

## Legend

 Event Trigger

 Objects

 Step Functions

 Textract

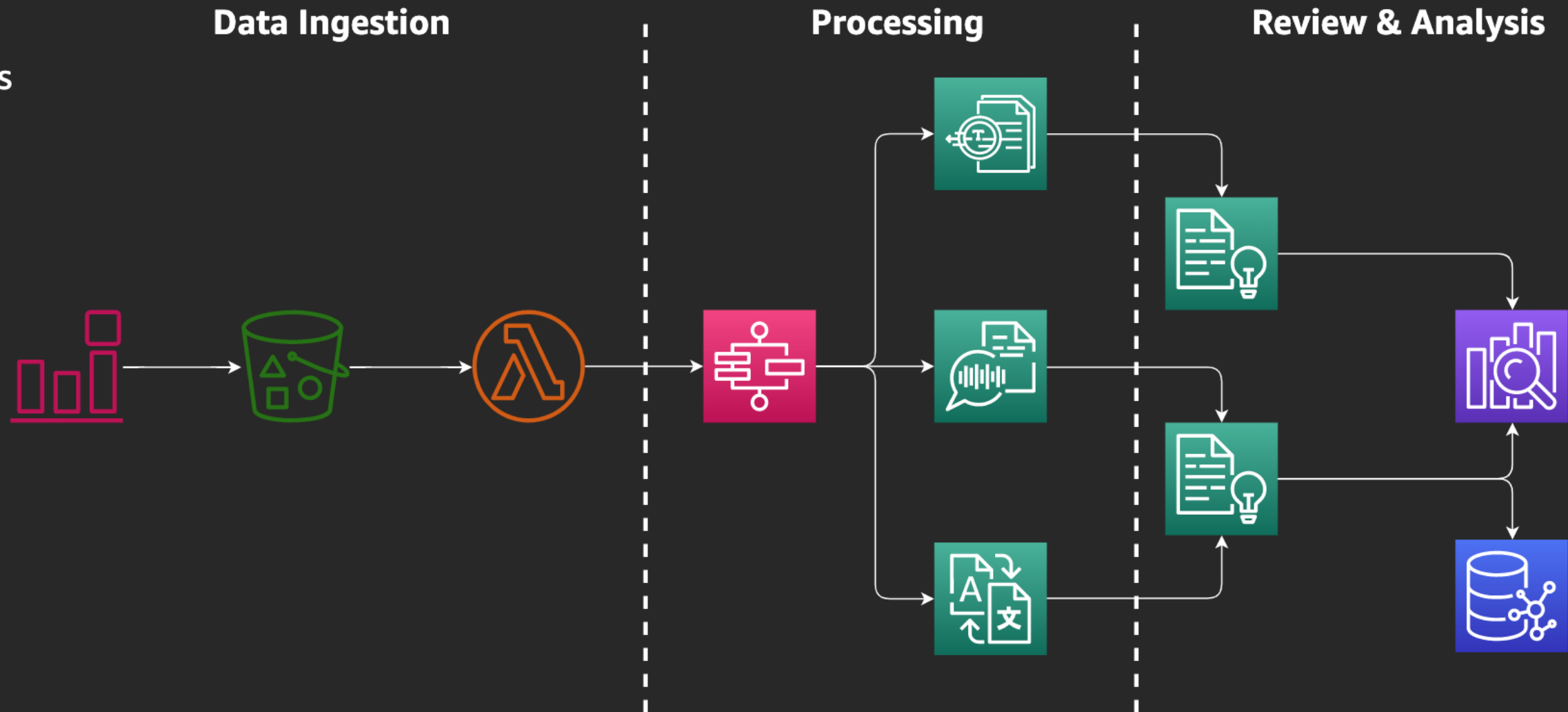
 Transcribe

 Translate

 Comprehend

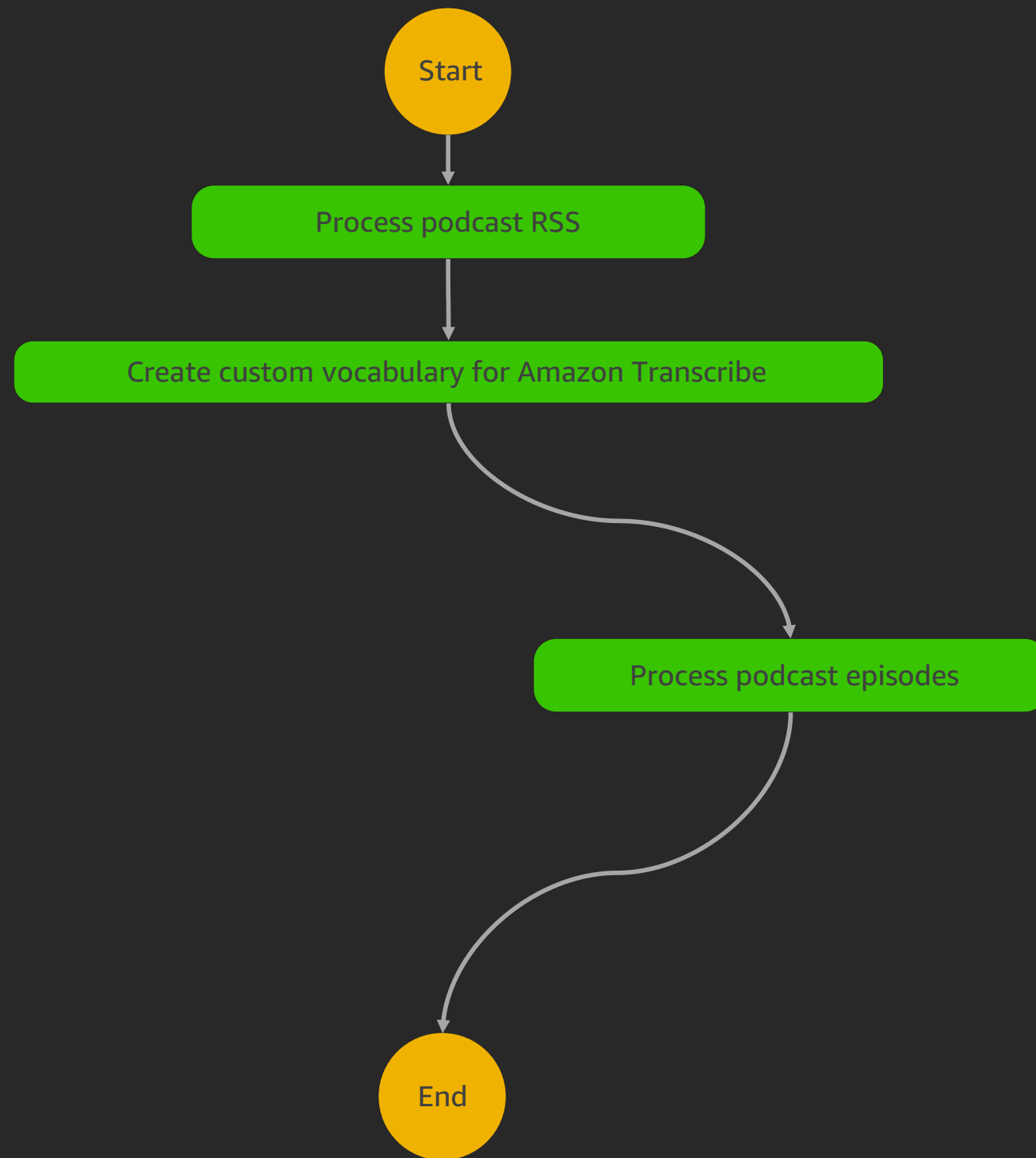
 Neptune

 Elastic Search





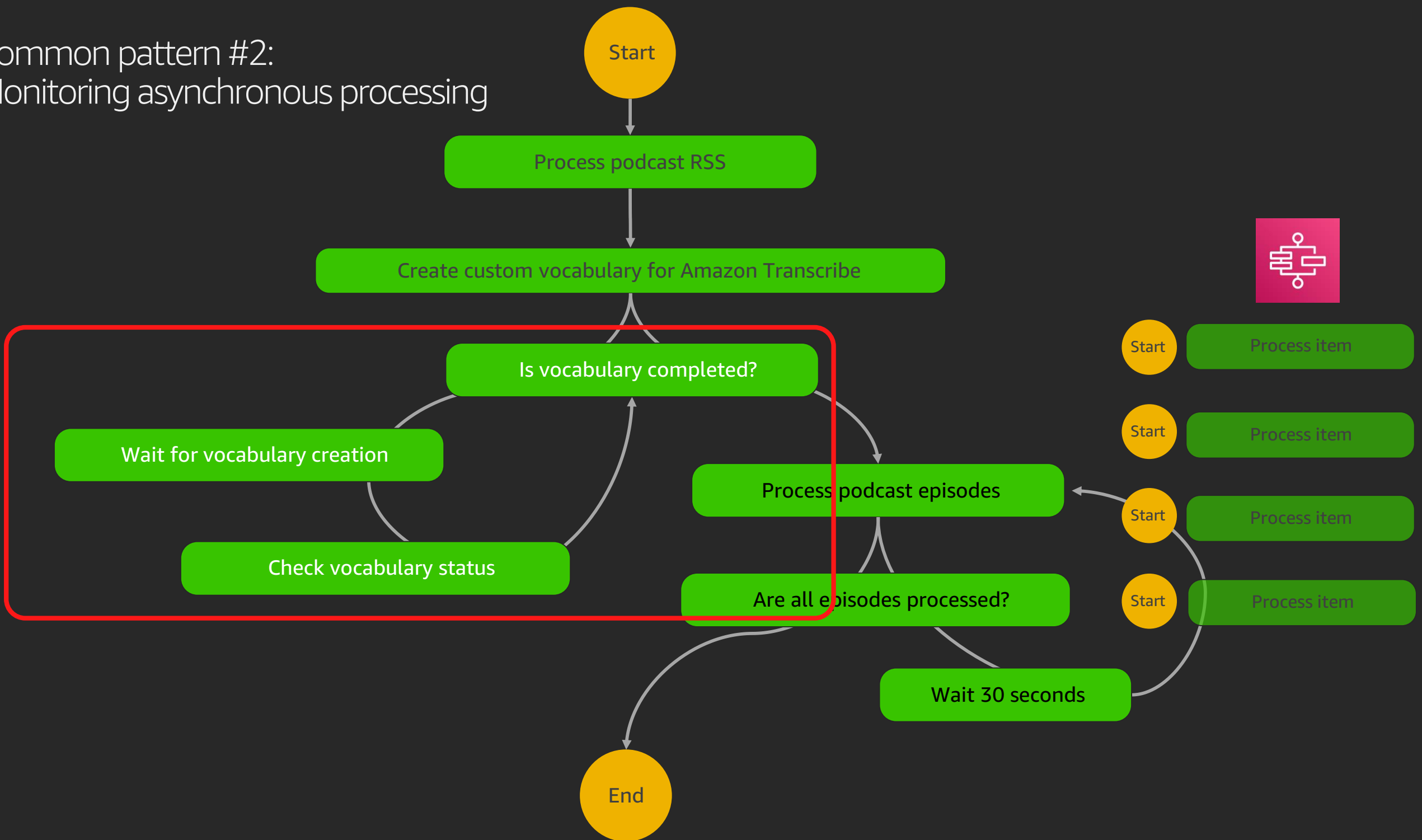
AWS Step  
Functions





AWS Step  
Functions

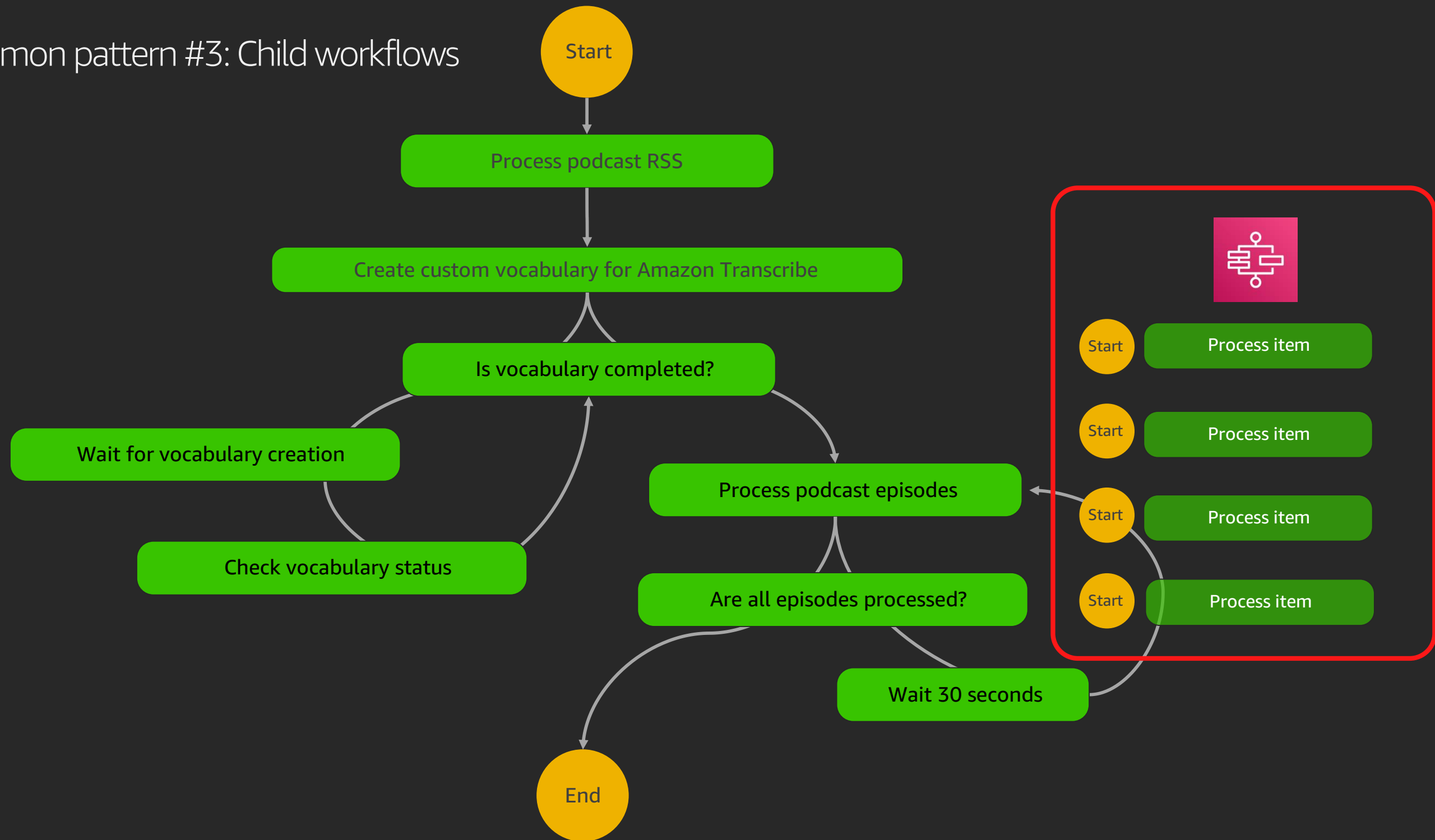
## Common pattern #2: Monitoring asynchronous processing





AWS Step  
Functions

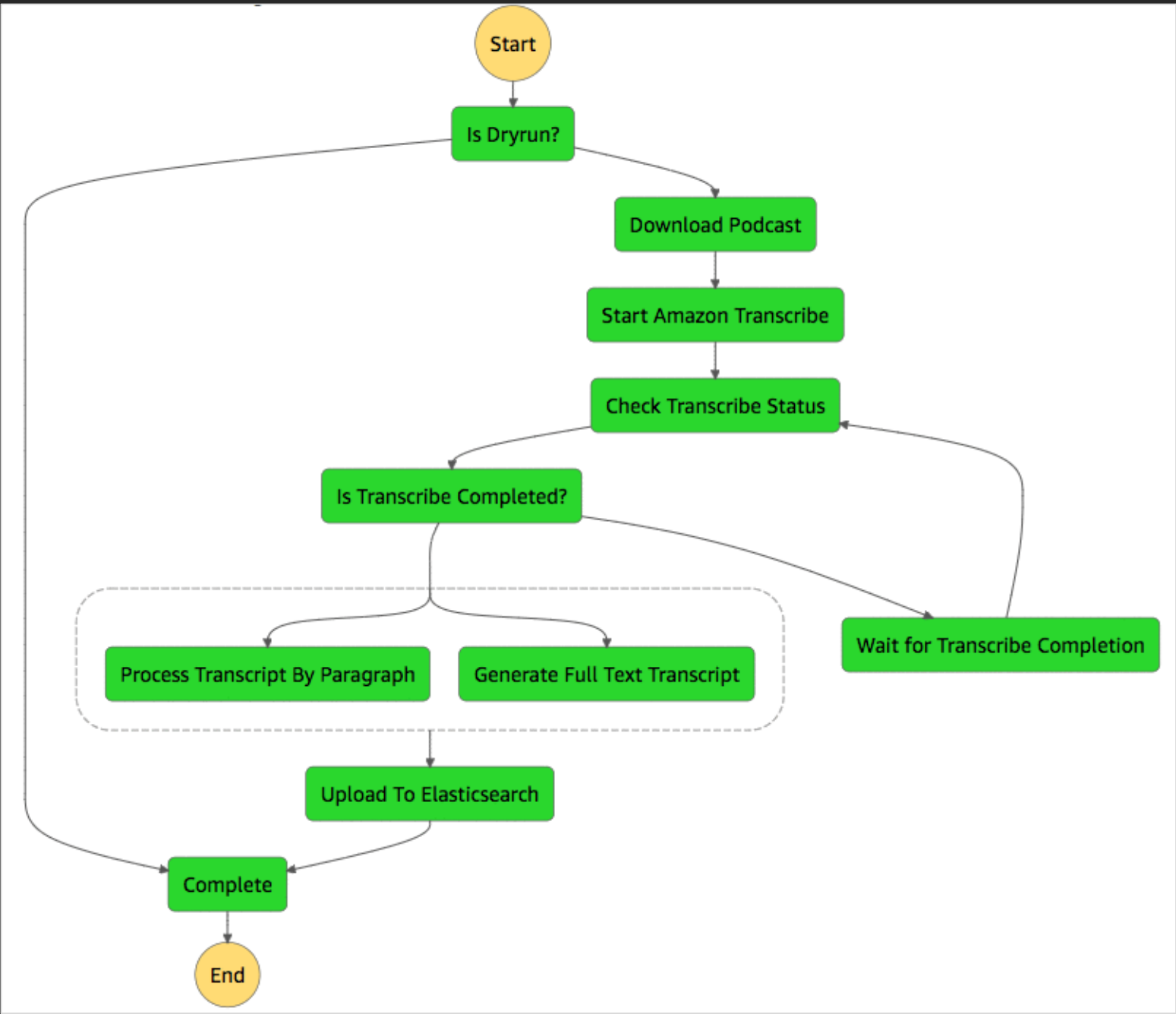
## Common pattern #3: Child workflows



# Common pattern #3: Child workflows



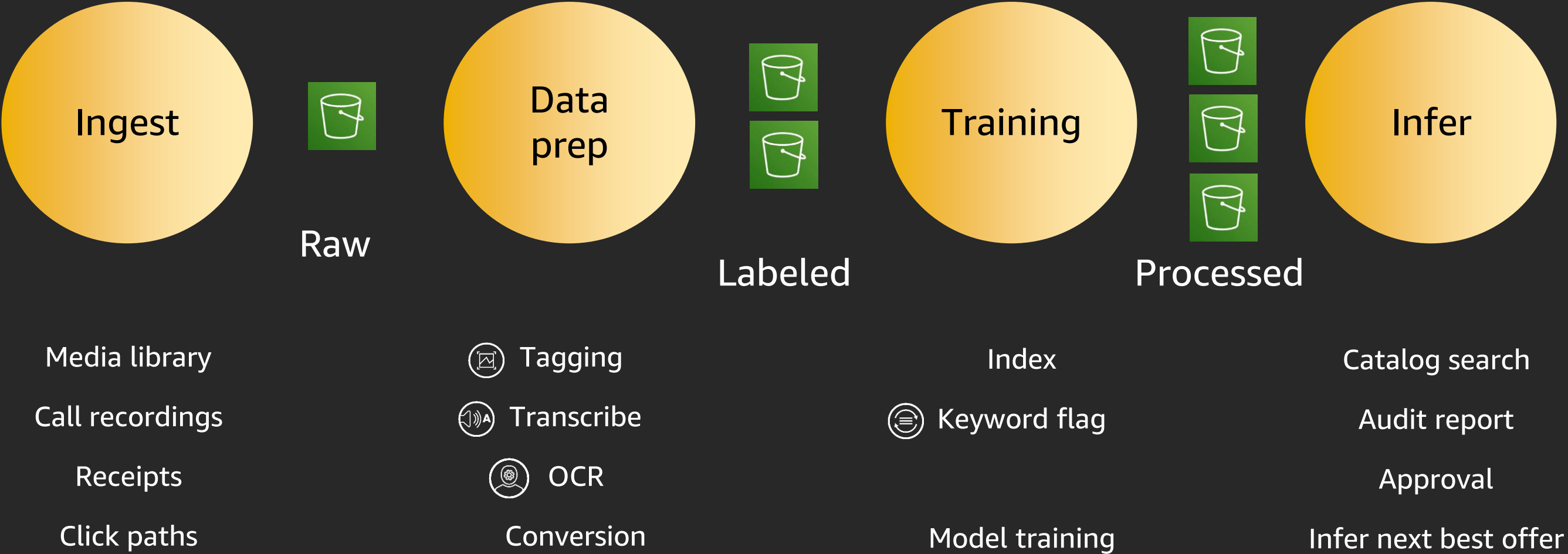
AWS Step  
Functions





# Demo

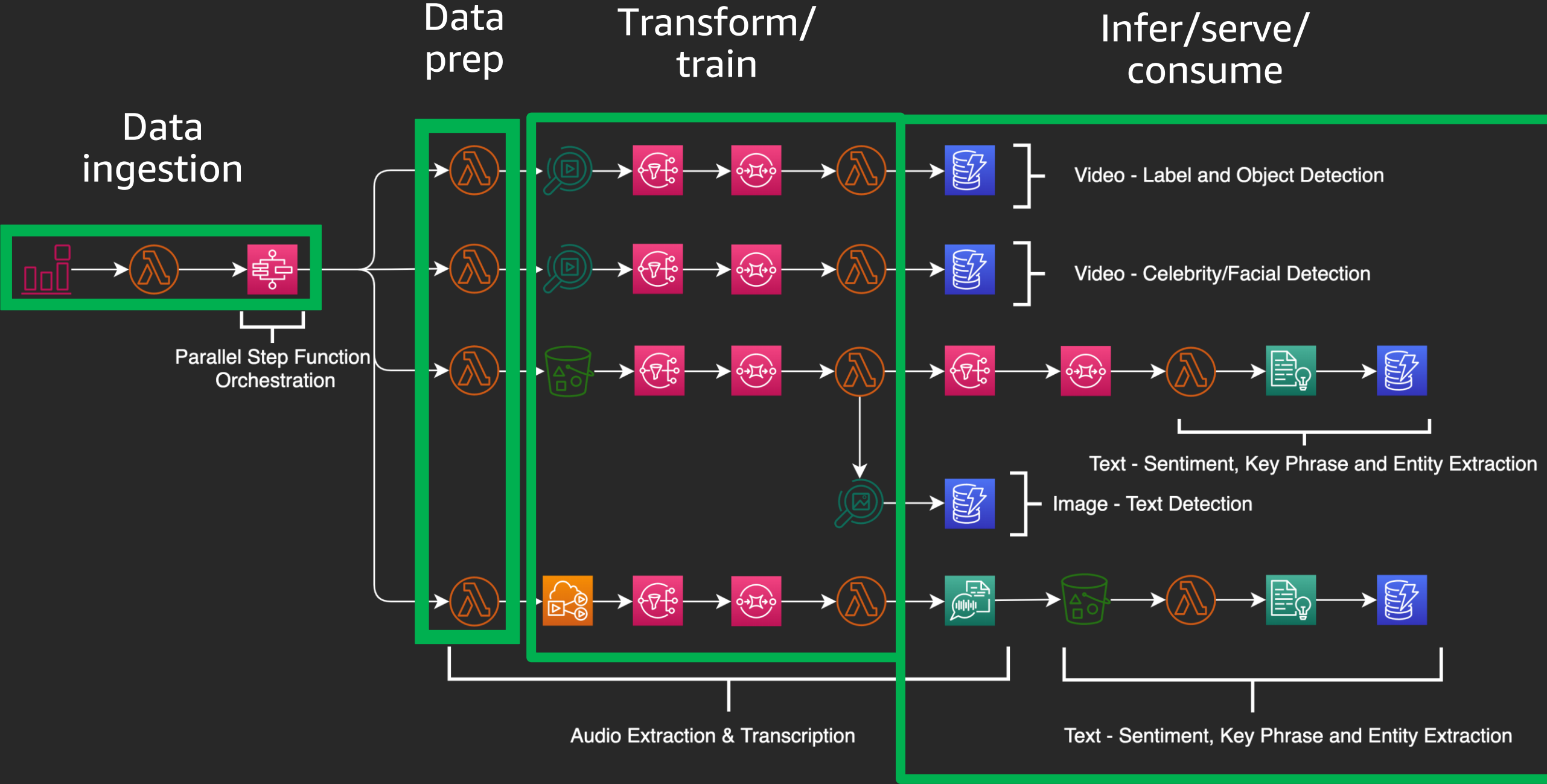
# Common pattern #4: Data available at all stages



# Common pattern #4: Data available at all stages

**Legend**

- Rekognition
- Rekognition - Image
- SNS
- SQS
- Elastic Transcoder
- DynamoDB
- Comprehend
- Transcribe
- Step Functions
- Event Trigger



# Discussion

# Thank you!

**Mark Roy**

roymark@amazon.com

**Ibrahim Gabr**

ibragabr@amazon.com



Please complete the session  
survey in the mobile app.