

# **Technical Description**

## **Lambda Logic & Control Flow**

**Group 2**

*GitHub Repository:*

[\*https://github.com/siddharth404/icc-aws-pdf-textract-project/\*](https://github.com/siddharth404/icc-aws-pdf-textract-project/)

## 1. Submission Lambda (Initiator)

Trigger: S3 ObjectCreated (incoming/\*.pdf)

Pseudocode Flow:

1. EXTRACT bucket, key FROM event record
2. VALIDATE file extension IS .pdf
3. CALL Textract.StartDocumentAnalysis(  
    DocumentLocation={Bucket, Key},  
    NotificationChannel={SNS\_Topic\_ARN, Role\_ARN}  
)
4. LOG JobId to CloudWatch
5. RETURN 200 OK

## 2. Processing Lambda (Worker)

Trigger: SQS Message (Batch Size: 1)

Pseudocode Flow:

1. PARSE Message Body -> SNS Notification -> JobId, Status
2. IF Status != 'SUCCEEDED':  
    MOVE SourceFile TO error/  
    RETURN
3. CALL Textract.GetDocumentAnalysis(JobId)
4. WHILE NextToken EXISTS:  
    CALL GetDocumentAnalysis(JobId, NextToken)  
    APPEND Blocks TO List
5. FOR Block IN Blocks:  
    IF Type == 'QUERY\_RESULT' AND Confidence > 90%:  
        MAP Answer TO Field (Name, Email, etc.)
6. GENERATE CSV String
7. WRITE CSV TO S3 (processed/)
8. COPY SourceFile TO S3 (archive/)
9. DELETE SourceFile FROM S3 (incoming/)

## 3. Resilience Strategies

1. \*\*Pagination Handling\*\*: The 'GetDocumentAnalysis' loop ensures large multi-page resumes are fully processed, not just the first page.
2. \*\*Confidence Filtering\*\*: Any field with <90% confidence is treated as NULL to maintain data integrity.

3. **\*\*Idempotency\*\***: S3 object moves (Copy+Delete) ensure files are strictly processed once per successful run.
4. **\*\*SQS Visibility Timeout\*\***: Set to 300s. If Lambda crashes, the message becomes visible again for retry.
5. **\*\*Dead Letter Queue (DLQ)\*\***: After 3 failed retries, messages move to DLQ for manual inspection.