Design Document

Secure Shield

PII Redaction Data Pipeline

Paris, 15th August 2025

Generated by:
Amazon Web Services EMEA SARL
(France Branch)

Customer:
HealthTech Analytics

Project member	Company	Role in the project
Miniar Jabri	AWS	Owner
Sadou-Bah Thierno	AWS	Mentor

Table of Contents

1. Executive Summary

1.1 Purpose

The Secure Shield PII Redaction Pipeline is designed to automate the detection and redaction of Personally Identifiable Information (PII) and Protected Health Information (PHI) from healthcare documents. This solution addresses the critical need for efficient, accurate, and compliant data processing in healthcare analytics environments.

Key Drivers:

- Current manual redaction process takes 30-45 minutes per document
- Human error rate of up to 7% poses compliance risks
- Increasing volume of healthcare documents requiring processing
- Compliance requirements for HIPAA and data privacy regulations
- Need for cost-effective scaling of data processing operations

1.2 Scope

In Scope:

- Automated document processing pipeline for healthcare documents
- Support for multiple document formats (PDF, TIFF, PNG, JPEG)
- PII/PHI detection using AWS Comprehend Medical
- Automated redaction with configurable rules
- Real-time processing status monitoring
- Integration with existing analytics workflows
- Security controls and HIPAA compliance measures
- User interface for document upload and management
- Reporting and audit capabilities

Out of Scope:

- Manual review workflow (future enhancement)
- Integration with legacy systems (separate project)
- Real-time streaming data processing
- Custom ML model development
- External API integrations (future phase)

1.3 Document Overview

This design document provides a comprehensive technical blueprint for implementing the Secure Shield solution. It covers:

Solution Architecture

- Serverless architecture leveraging AWS services
- End-to-end workflow design
- Security and compliance controls
- Performance optimization strategies

Operational Considerations

- Monitoring and alerting framework
- Cost optimization approaches

Target Audience

This document is intended for:

- Technical implementation teams
- Solution architects
- Security reviewers
- Operations teams
- Project stakeholders

Success Criteria

The implementation will be considered successful when:

- Processing time reduced to <5 minutes per document
- Accuracy rate achieves >99%
- System scales to handle 5000+ documents per day
- Meets all HIPAA compliance requirements
- Achieves cost reduction targets compared to manual processing

2.Business Context

2.1 Customer Overview

HealthTech Analytics is a healthcare data analytics company that processes large volumes of patient records and healthcare data to derive insights for improving patient care and operational efficiency. They currently handle sensitive patient information and need to protect it while maintaining data utility for analytics purposes.

2.2 Business Problems and Opportunities

- Problem: Manual redaction of PII/PHI is time-consuming (30-45 minutes per document) and error-prone (up to 7% error rate).
- Problem: Compliance violations due to human error can result in penalties of up to $50,000 per incident.
- Problem: Current manual process doesn't scale with increasing document volumes, causing processing delays.
- Opportunity: Automating the redaction process can significantly reduce processing time and improve accuracy.
- Opportunity: Implementing a scalable solution can handle growing data volumes and support business growth.
- Opportunity: Improving compliance can enhance reputation and trust with clients and regulators.

2.3 Key Performance Indicators

- Document processing time: Target <5 minutes per document
- Accuracy rate: Target >99% for PII/PHI detection and redaction
- Scalability: Ability to process 5000+ documents per day
- Compliance: Zero HIPAA violations related to data handling
- Cost reduction: 70% reduction in operational costs compared to manual processing

3.Requirements

3.1 Functional Requirements

- Automated ingestion of multiple document formats (PDF, TIFF, PNG, JPEG)
- Text extraction while preserving document structure
- PII/PHI detection using AWS Comprehend Medical with customizable entity types
- Configurable redaction rules with different strategies (full redaction, partial masking)
- Real-time processing status monitoring
- Secure storage of original and redacted documents
- User interface for document upload and management
- Integration with existing analytics workflows
- Comprehensive audit logging and reporting capabilities

3.2 Non-functional Requirements

- Performance: Process small documents (<5MB) in <2 minutes, medium documents (5-20MB) in <5 minutes, large documents (20-100MB) in <15 minutes
- Scalability: Handle peak processing of 1000 documents/hour, sustained processing of 500 documents/hour

- Reliability: 99.99% uptime for the redaction service
- Security: End-to-end encryption for data at rest and in transit
- Compliance: Full HIPAA compliance for data handling and storage
- Usability: Intuitive user interface with minimal training required for operators
- Maintainability: Comprehensive logging and monitoring for easy troubleshooting

3.3 Technical Tenets

- The solution values serverless architecture over traditional server-based deployments to improve scalability and reduce operational overhead.
- The solution values automation over manual processes to increase efficiency and reduce human error.
- The solution values modular design over monolithic architecture to enhance maintainability and enable future enhancements.
- The solution values cloud-native services over custom-built components to leverage AWS expertise and reduce development time.
- The solution values strong access controls and encryption over complex network segmentation to ensure data security while maintaining flexibility.

4. Project Overview

4.1 Objective

Create a serverless pipeline to automatically detect and redact PII from various healthcare document types, ensuring data privacy and HIPAA compliance while preserving document structure for analysis.

4.2 Stakeholders

- HealthTech Analytics (Client)
- AWS Professional Services Team
- Healthcare data analysts and researchers

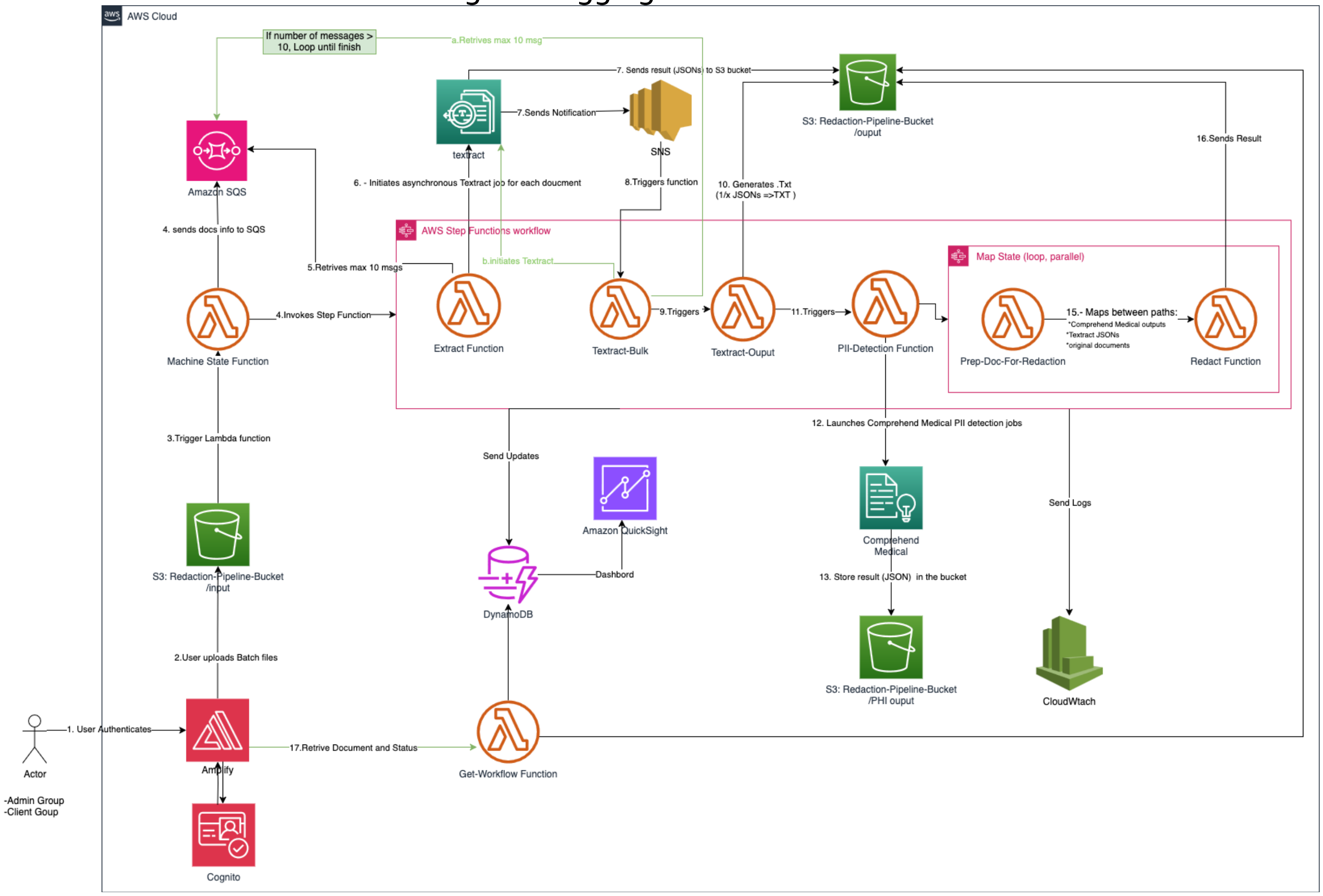
5. HLD: High Level Architecture and Diagram:

The solution is built on a serverless architecture using AWS services:

5.1 Core Components

- Amazon S3: Document storage
- AWS Lambda: Serverless compute for processing
- Amazon Textract: Document text extraction
- Amazon Comprehend Medical: Healthcare-specific PII detection

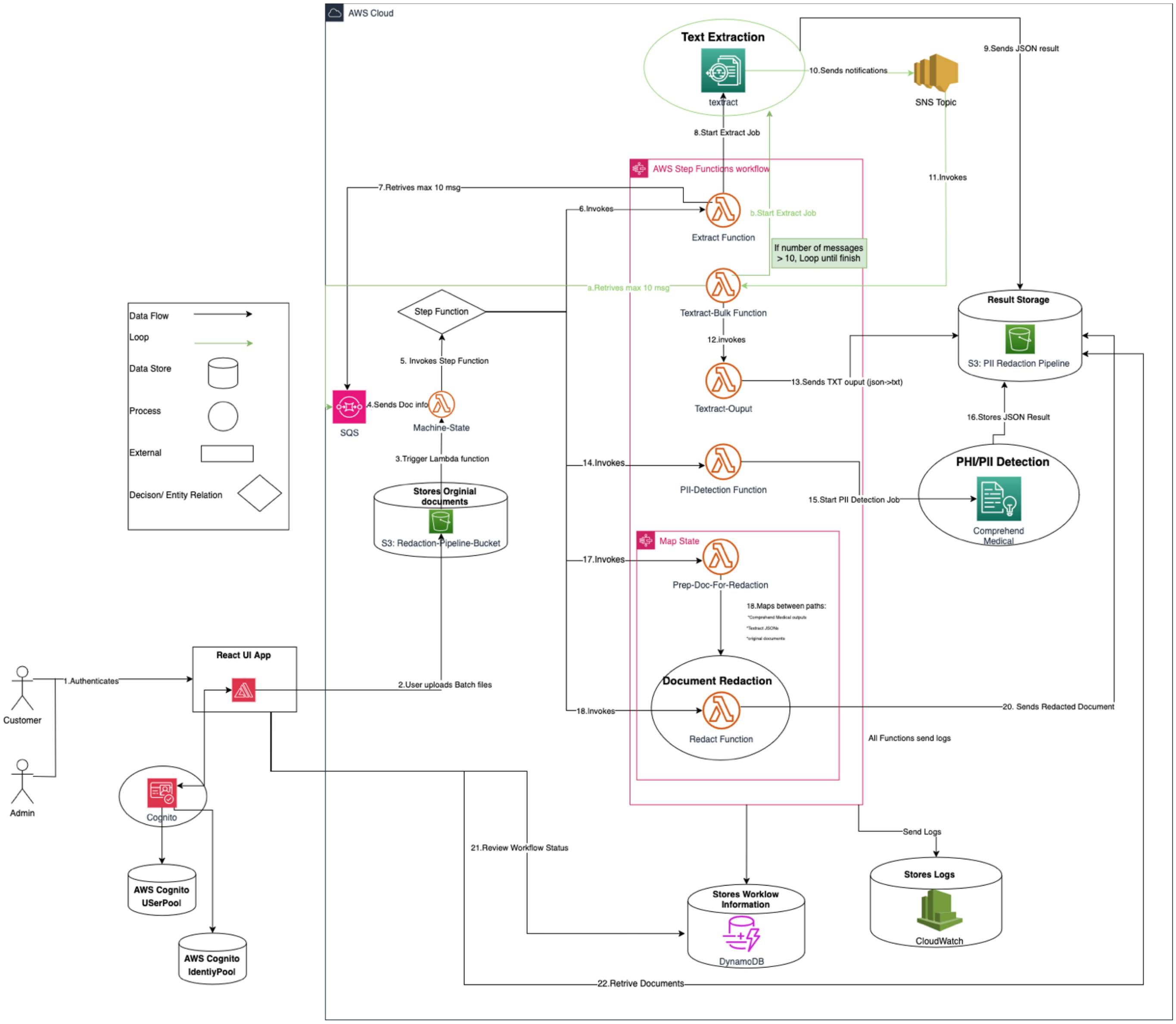
- AWS Step Functions: Workflow orchestration
- Amazon DynamoDB: Metadata and job status tracking
- Amazon Cognito: User authentication
- Amazon CloudWatch: Monitoring and logging


AWS Cloud
If number of messages > 10, Loop until finish
a.Retrives max 10 msg
7. Sends result (JSONs) to S3 bucket
7.Sends Notification
textract
SNS
Amazon SQS
6. - Initiates asynchronous Textract job for each doucment
8.Triggers function
10. Generates .Txt (1/x JSONs =>TXT)
S3: Redaction-Pipeline-Bucket /ouput
16.Sends Result
4. sends docs info to SQS
AWS Step Functions workflow
b.initiates Textract
5.Retrives max 10 msgs
Map State (loop, parallel)
4.Invokes Step Function
9.Triggers
11.Triggers
15.- Maps between paths:
*Comprehend Medical outputs
*Textract JSONs
*original documents
Machine State Function
Extract Function
Textract-Bulk
Textract-Ouput
PII-Detection Function
Prep-Doc-For-Redaction
Redact Function
3.Trigger Lambda function
Send Updates
12. Launches Comprehend Medical PII detection jobs
Send Logs
Amazon QuickSight
Comprehend Medical
S3: Redaction-Pipeline-Bucket /input
Dashbord
13. Store result (JSON) in the bucket
DynamoDB
2.User uploads Batch files
S3: Redaction-Pipeline-Bucket /PHI ouput
CloudWtach
1. User Authenticates
Actor
17.Retrive Document and Status
Amplify
Get-Workflow Function
-Admin Group
-Client Goup
Cognito

5.2 High-Level Workflow

1. User uploads documents to S3 via a secure web interface
2. S3 event triggers a Step Functions workflow
3. Textract extracts text from documents
4. Comprehend Medical identifies PII entities
5. Lambda functions process and redact identified PII
6. Redacted documents are stored in S3
7. Job status and metadata are updated in DynamoDB
8. Users can monitor progress and retrieve redacted documents

6. Data Flow



6.1 Input

- Healthcare documents (PDF, TIFF, PNG, JPEG)

6.2 Processing

- Text extraction results from Textract
- PII entity detection results from Comprehend Medical
- Redaction metadata (e.g., number of entities redacted, confidence scores)

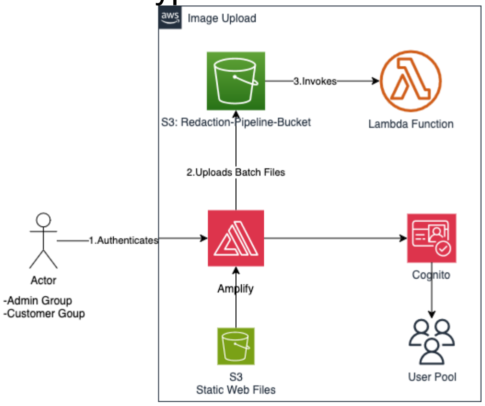
6.3 Output

- Redacted documents in original format
- Processing metadata and job status information

7. Detailed Design

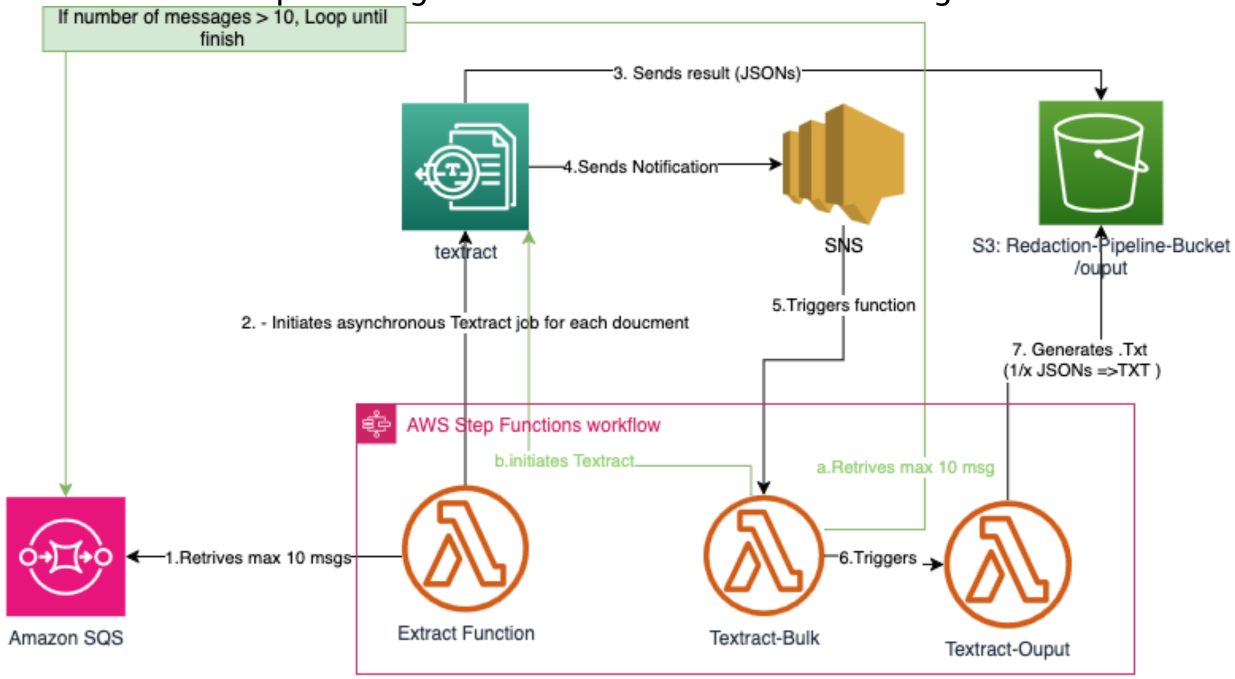
7.1 Document Ingestion

- Use Amazon Cognito for user authentication
- Implement a React-based web interface for document upload
- Store original documents in an encrypted S3 bucket



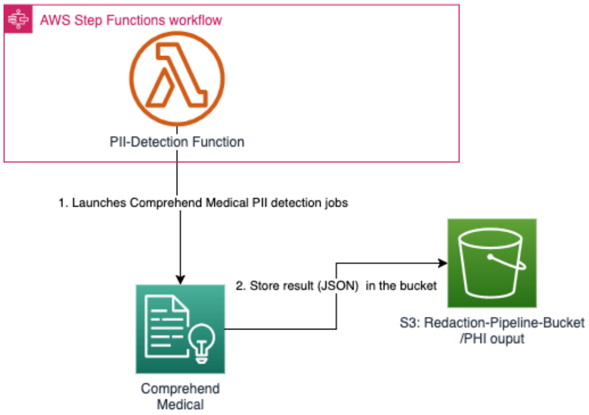
7.2 Text Extraction

- Use Textract's asynchronous API for batch processing
- Extract text while preserving document structure and formatting


If number of messages > 10, Loop until finish
3. Sends result (JSONs)
textract
4.Sends Notification
SNS
S3: Redaction-Pipeline-Bucket /ouput
2. - Initiates asynchronous Textract job for each doucment
5.Triggers function
7. Generates .Txt (1/x JSONs =>TXT)
AWS Step Functions workflow
b.initiates Textract
a.Retrives max 10 msg
Amazon SQS
1.Retrives max 10 msgs
Extract Function
Textract-Bulk
6.Triggers
Textract-Ouput

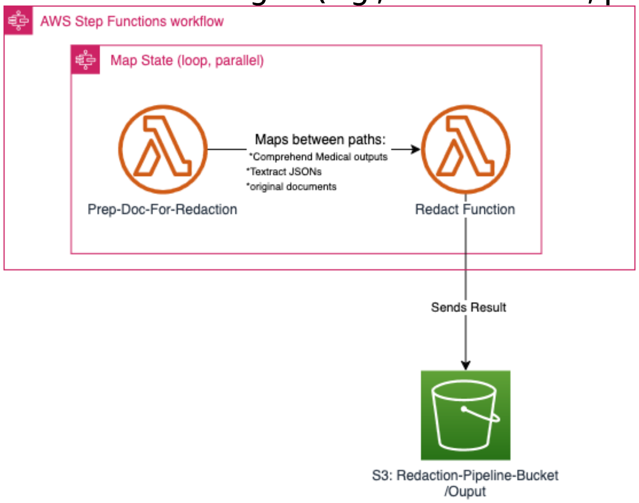
7.3 PII Detection

- Utilize Comprehend Medical for healthcare-specific entity recognition
- Configure custom entity types for additional PII categories
- Implement confidence thresholds for entity detection


AWS Step Functions workflow
PII-Detection Function
1. Launches Comprehend Medical PII detection jobs
2. Store result (JSON) in the bucket
Comprehend Medical
S3: Redaction-Pipeline-Bucket /PHI ouput

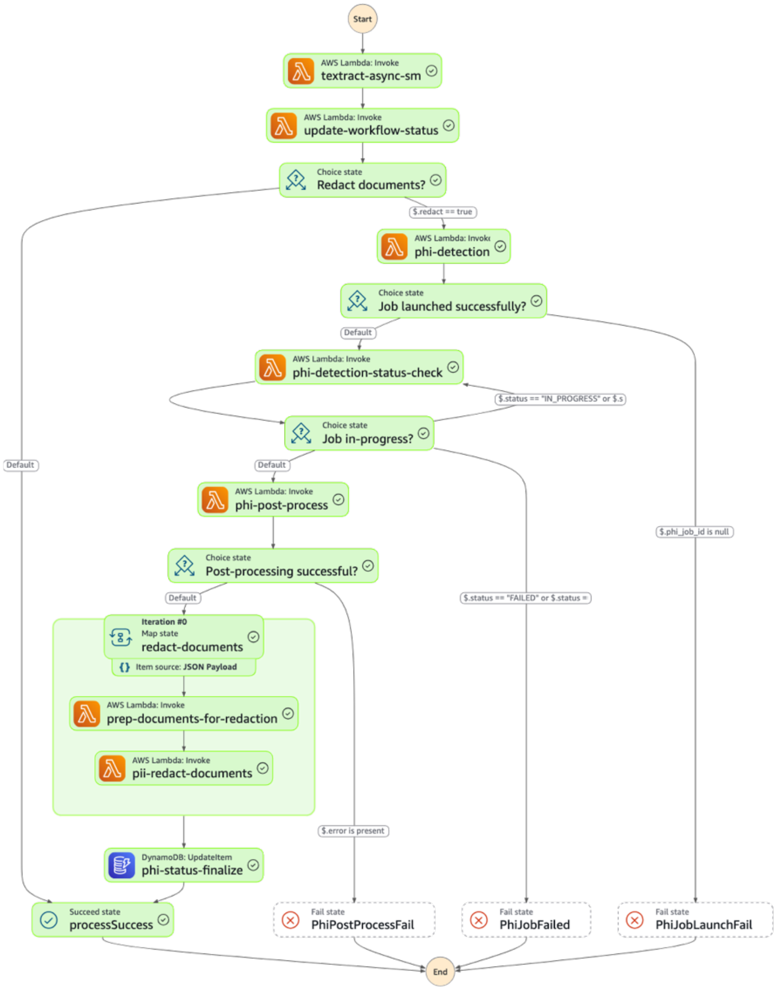
7.4 Redaction Process

- Develop Lambda functions to apply redactions based on detected entities
- Preserve document structure by replacing PII with placeholders
- Implement different redaction strategies (e.g., full redaction, partial masking)


AWS Step Functions workflow
Map State (loop, parallel)
Prep-Doc-For-Redaction
Maps between paths:
*Comprehend Medical outputs
*Textract JSONs
*original documents
Redact Function
Sends Result
S3: Redaction-Pipeline-Bucket /Ouput

7.5 Workflow Orchestration

- Design a Step Functions state machine to manage the end-to-end process
- Implement error handling and retry logic for each step
- Use DynamoDB to track job status and metadata



7.6 Security and Compliance

- Encrypt data at rest and in transit using AWS KMS
- Implement fine-grained IAM roles and policies
- Set up CloudTrail for comprehensive auditing
- Configure VPC endpoints for enhanced network security

7.7 Monitoring and Logging

- Set up CloudWatch for real-time monitoring
- Configure alarms for error conditions and performance thresholds
- Implement detailed logging for all processing steps
- Set up QuickSight Dashbord for real-time monitoring

8. Scalability and Performance

8.1 Serverless Architecture Scaling

- AWS Lambda Auto-scaling Configuration

- Initial concurrency: 100 concurrent executions
- Maximum concurrency: 1000 concurrent executions
- Reserved concurrency: 50 for critical functions
- Scaling triggers: CPU utilization > 70%, Memory usage > 80%

8.2 Batch Processing Configuration

- Document Batch Specifications
- Maximum batch size: 100 documents
- Maximum batch payload: 500MB
- Processing window: 20 minutes per batch
- Concurrent batch limit: 10

8.3 Lambda Function Optimization

- Resource Allocation
- Document Processor: 1024MB memory, 5-minute timeout
- PII Detector: 2048MB memory, 10-minute timeout
- Redaction Handler: 1024MB memory, 15-minute timeout

8.4 Performance Metrics

- Target SLAs • Small documents (<5MB): < 2 minutes processing time
- Medium documents (5-20MB): < 5 minutes processing time
- Large documents (20-100MB): < 15 minutes processing time
- Throughput Targets
- Peak processing: 1000 documents/hour
- Sustained processing: 500 documents/hour
- Batch processing: 5000 documents/day

9. Cost Optimization

9.1 S3 Storage Optimization

- Intelligent-Tiering Configuration
- Monitoring and automation charges: $0.0025 per 1,000 objects
- Automatic tiering between:
  - Frequent Access tier (accessed monthly)
  - Infrequent Access tier (not accessed for 30 days)
  - Archive Instant Access tier (not accessed for 90 days)

9.2Lifecycle Management

- Document Retention Policy : 0-90 days: Glacier storage

- Deletion Rules
  - Incomplete multipart uploads: Delete after 7 days
  - Processing temporary files: Delete after 24 hours
  - Failed processing artifacts: Delete after 30 days

9.3 Lambda Optimization

- Memory Optimization:
  - Regular performance profiling
  - Automated memory adjustment based on execution metrics
  - Cold start optimization using Provisioned Concurrency
- Cost Monitoring
  - Per-function cost tracking using cost allocation tags
  - Monthly cost analysis and optimization reviews
  - Automated cost anomaly detection

10. Future Enhancements

10.1 Custom PII Detection Models

- Model Development
- Training data requirements: 10,000+ annotated documents
- Custom entity types specific to healthcare domain
- Model validation and testing framework
- Integration Plan
- API development for model deployment
- A/B testing framework for accuracy comparison
- Performance monitoring and optimization

10.2 Redaction Management Interface

- Features
  - Custom redaction rule creation
  - Entity type management
  - Confidence threshold adjustment
  - Batch processing configuration
  - Audit logging and reporting
- User Access Levels:
  - Administrator: Full access to all settings
  - Supervisor: Rule management and monitoring
  - Analyst: View-only access to rules and statistics

11. Deployment and Operations

11.1 Infrastructure as Code

- AWS CDK Implementation • Stack Organization:
  - Network Stack (VPC, Subnets)
  - Security Stack (IAM, KMS)
  - Application Stack (Lambda, S3)
  - Monitoring Stack (CloudWatch) • Environment Management:
  - Development
  - Staging
  - Production • Version Control:
  - Git repository structure
  - Branch strategy
  - Code review process

11.2 Operational Runbooks

- Common Operations: • System Startup/Shutdown • Backup/Restore Procedures • Scaling Operations • Emergency Procedures
- Troubleshooting Guides: • Error Resolution Steps • Performance Issues • Security Incidents • Service Disruptions

12. Timeline and Milestones

Project Team:
• 1 ProServe intern (100% allocation)
- Project owner
- Solution development
- Documentation
- Testing and implementation

Support Team:
• 1 Mentor (20% allocation)
- Technical guidance
- Code reviews
- Architecture validation
- Weekly 1:1s

• 1 Manager (10% allocation)
- Project oversight
- Career development
- Weekly team meetings
- Performance feedback

Phase 1: Research and Design (2 weeks)

Week 1:

- Project requirements gathering
- AWS services research
- Architecture design

Deliverables:

- Project proposal
- Architecture diagram
- Initial design document

Week 2:

- Component design
- Security planning
- Mentor review and feedback

Deliverables:

- Detailed design document
- Security framework
- Project timeline

Phase 2: Development and Testing (4 weeks)

Week 3-4:

- Frontend development (React)
- Backend setup (Lambda, Step Functions)
- Initial integration

Deliverables:

- Working prototype
- Code repository setup
- Unit tests

Week 5-6:

- Integration completion
- Security implementation
- Performance optimization

Deliverables:

- Functional application
- Test cases
- Security documentation

Phase 3: Documentation and Presentation (2 weeks)

Week 7:

- User guide creation
- Technical documentation
- Bug fixes and improvements

Deliverables:

- User documentation
- Technical documentation
- Final testing results

Week 8:
- Presentation preparation
- Demo refinement
- Knowledge transfer

Deliverables:
- Final presentation
- Project handover
- Live demo

Weekly Touchpoints:

- Weekly 1:1 with mentor (30 mins)
- Bi-weekly review with manager (30 mins)

Key Learning Objectives:

1. AWS Services implementation
2. Security best practices
3. Infrastructure as Code
4. CI/CD pipelines
5. Documentation skills
6. Presentation skills

Success Metrics:

- Functional solution deployment
- Documentation completion
- Successful final presentation
- Code quality standards met
- Security requirements fulfilled

Total Project Duration: 8 weeks

13. Conclusion

This design document outlines a comprehensive solution for automated PII redaction in healthcare documents using AWS services. The serverless architecture provides a scalable, secure, and cost-effective approach to handling sensitive data while meeting HIPAA compliance requirements.