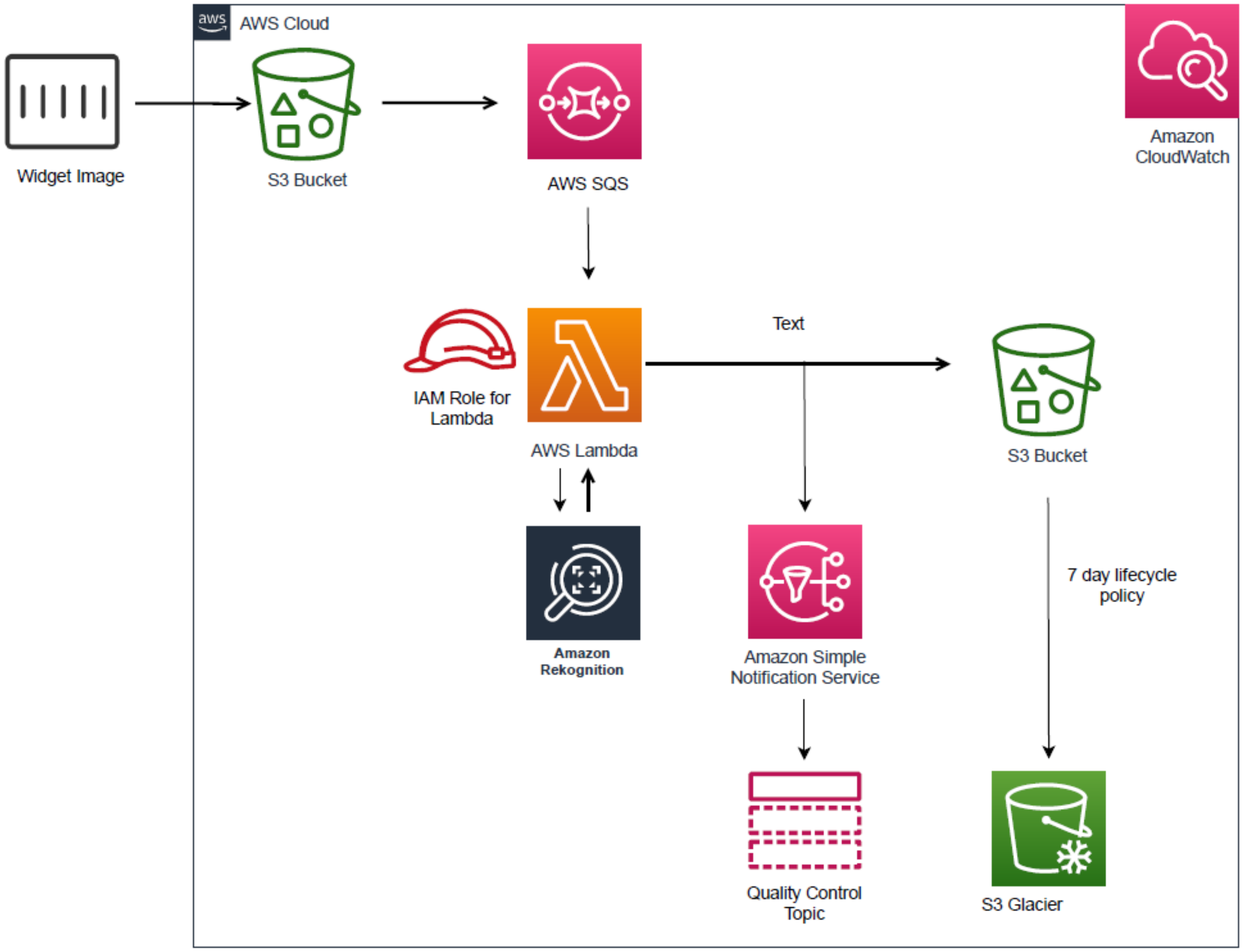
A. Summary of Design, Problem, and Approach



Our serverless architecture intends to automate an image validation and message sending workflow that is currently done by hand. Our collective approach to the design is in response to the specific business requirement of 100% automation, which is the primary goal that the company is hoping to achieve. Our design therefore leverages AWS Lambda and AWS Simple Notification Service to assist with event-based automation and message sending, the two most critical components of the solution. The rest of the architecture fulfills standard necessities such as storage, monitoring and logging, and image validation based on machine learning. The design ultimately emphasizes serverless design, scalability, and full automation while keeping costs below our prescribed limit.

B. Description

Our automation process begins with the picture being uploaded to an S3 bucket. As a simple but reliable storage service, it provides a place for cameras to upload their photos to to begin the workflow. Having a dedicated bucket for initial image upload also simplifies the IAM permissions for the assembly line cameras, which need access to nothing else otherwise. S3 has 11 9's of durability and is highly scalable and secure, which makes it a great choice for the initial storage.

The next step in the process is AWS SQS, a standard queue in which items uploaded to the first step buckets are automatically placed in. The purpose of the standard queue as opposed to a FIFO queue is to guarantee at-least-once-delivery, which ensures that in the event of a large-scale failure, no widget tasks and lost and can continue to be processed by the lambda function with minimal business impact. This helps improve the overall robustness of the service and prevents problematic data loss.

Next up, AWS Lambda is triggered by the addition of the image to the SQS queue and begins the serverless image validation process. Lambda is a natural choice for this step since it allows the user to run code without a physical server. By allowing us to run our own code, we're able to leverage this function to help run the image through Amazon Rekognition, report the results to Quality control, and store the results in a second S3 bucket. It's also highly scalable within a region and has the added benefit of being able to run code in parallel and respond to each trigger independently, effectively allowing the lambda function to accommodate any scale of traffic.

Another crucial step of the process is AWS Rekognition. This service uses machine learning to analyze the widget and is therefore the key step to automating a process that would otherwise have to be done manually by the inspector. Rekognition is also scalable, fully managed, and easily accessible even for workers with no knowledge of machine learning, which can help the company continue to save costs on hiring specific domain knowledge.

AWS SNS and its quality control topic serve the purpose of notifying quality control of the results of the image validation. This is a scalable solution that allows Quality Control to continue receiving messages with reports of the analyses. SNS is also fully automated and replaces the need to have an individual manually send messages. SNS has the added benefit of allowing different people and groups to subscribe to the topic, allowing users to be added to and removed from the message list without interrupting the workflow.

A second S3 bucket is needed at this point in the workflow since we need a place to store the results of the image processing. Note that this is a separate bucket from the one that assembly line cameras have access to and therefore will follow a different set of permissions. We elected to use a second bucket instead of sending it automatically to long term storage in case complications arise and a particular result needs to be manually reexamined within 7 days. We believe this helps account for the practical necessity of preparing for possible issues with the workflow, allowing workers to grab the results immediately they move on to long term storage via lifecycle policy, where the retrieval process may take longer.

The final destination for images without issues is AWS Glacier. The business requirements state that they want the records to be available for 3 years, which glacier is well suited for. Glacier is optimized for long-term storage and therefore helps to save significantly on costs as opposed to storing every single widget photo from the last 3 years in an S3 bucket. Images will still be retrievable during this time period albeit not immediately, which is sufficient for the purpose of keeping records that are unlikely to require urgent access.

Additionally, we use CloudWatch for monitoring the entire workflow and collecting information on performance metrics. This has many benefits such as helping ensure that the load remains tolerable for each component in the overall process and that we process as many images we expect to. These metrics can ultimately help us understand which parts of our design face the highest workload and can allow us to make intelligent decisions to improve operational performance.

Security is managed with IAM roles for each moving part in the process. For example, assembly line cameras will be able to upload photos to the buckets that begin the overall process but will not be allowed to access the bucket that stores the results of the image validation. Each component will have appropriate IAM roles to ensure that data moves in an orderly fashion on top of being consistent with good design principles.

C. Design Considerations

There were a few major design considerations to consider while making the architecture. One of the most important ones was the ability to remain under a certain cost, as our task is considered failed if we exceed budget. This informed our decision to use tools such as AWS Glacier, which can help keep operational costs low given the use case. Another decision we made regarding budget is to