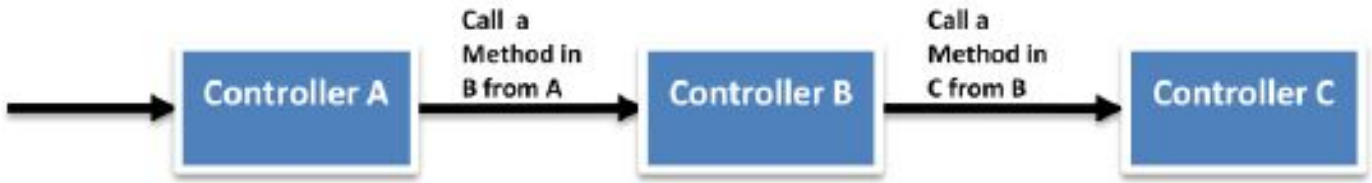
Amazon
Simple Queue Service
(SQS)

Sep 2017

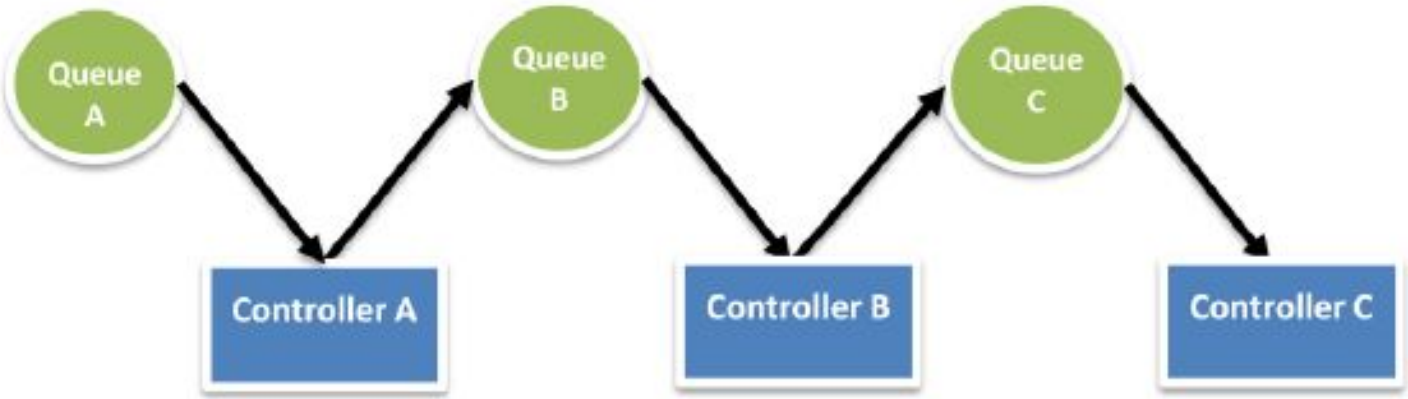
What is Amazon SQS ?

- **Hosted queue** for storing messages as they travel between applications or microservice
- Highly Available, Fail Safe , Distributed queue system
- Temporary repository for messages
- Buffer between PRODUCER & CONSUMER

Loosely Coupled Architecture with Q

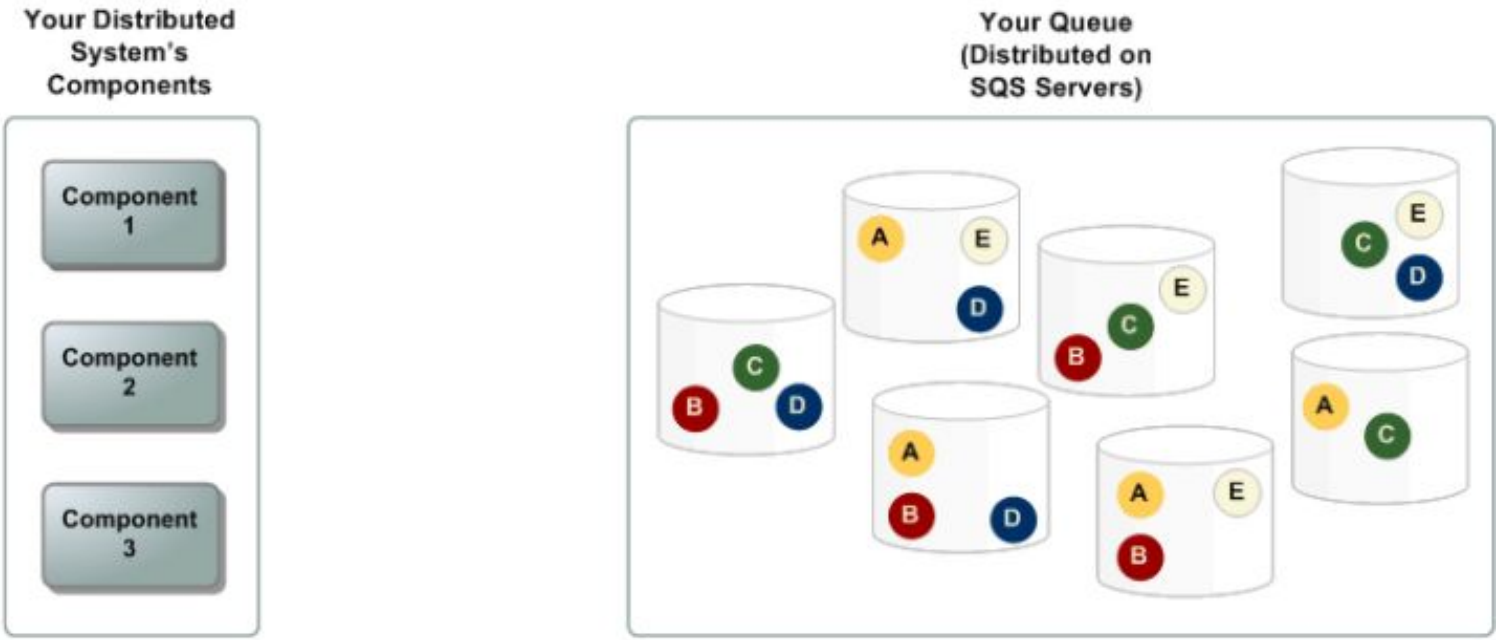


Tight coupling (procedural programming)



Loose coupling (independent phases using queues)

Architecture of SQS



- Messages are redundantly store across SQS servers

SQS Q - Types

- Standard Q
- FIFO Q (Early 2017)

Comparing Q

Standard	FIFO
Available in all regions. **High Throughput** – Standard queues can support a nearly unlimited number of transactions per second (TPS) per API action. **At-Least-Once Delivery** – A message is delivered at least once, but occasionally more than one copy of a message is delivered. **Best-Effort Ordering** – Occasionally, messages might be delivered in an order different from which they were sent.	Available in the US East (N. Virginia), US East (Ohio), US West (Oregon), and EU (Ireland) regions. **First-In-First-Out Delivery** – The order in which messages are sent and received is strictly preserved. **Exactly-Once Processing** – A message is delivered once and remains available until a consumer processes and deletes it. Duplicates are not introduced into the queue. **Limited Throughput** – Without batching, FIFO queues can support up to 300 messages per second (300 send, receive, or delete operations per second). If you take advantage of the maximum batching of 10 messages per operation, FIFO queues can support up to 3,000 messages per second.

Comparing Q

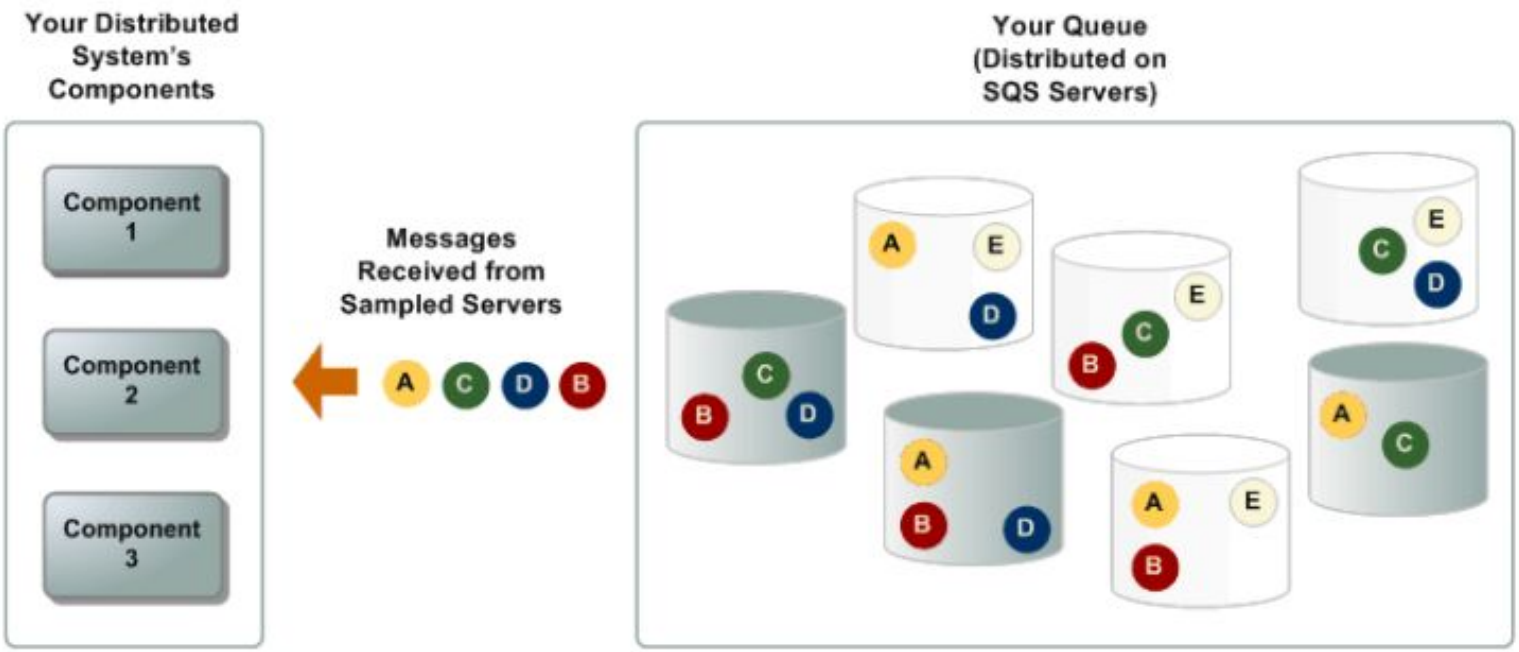
Standard	FIFO
	
Send data between applications when the throughput is important, for example: • Decouple live user requests from intensive background work: let users upload media while resizing or encoding it. • Allocate tasks to multiple worker nodes: process a high number of credit card validation requests. • Batch messages for future processing: schedule multiple entries to be added to a database.	Send data between applications when the order of events is important, for example: • Ensure that user-entered commands are executed in the right order. • Display the correct product price by sending price modifications in the right order. • Prevent a student from enrolling in a course before registering for an account.

Key Concepts

At least Once Delivery - Standard Q

- Amazon SQS stores copies of your messages on multiple servers for redundancy and high availability. On rare occasions, one of the servers that stores a copy of a message might be unavailable when you receive or delete a message.
- If this occurs, the copy of the message isn't deleted on that unavailable server, and you might get that message copy again when you receive messages.
- **Design your applications to be *idempotent* (they should not be affected adversely when processing the same message more than once)**

Short Polling



- Amazon SQS samples several of the servers (in gray) and returns the messages from those servers (Message A, C, D, and B).
- Message E isn't returned to this particular request, but is returned to a subsequent request.

Long Polling

- Long polling reduces the number of empty responses by allowing Amazon SQS to wait until a message is available in the queue before sending a response.
- Queries all (rather than a limited number) of the servers.
- Unless the connection times out, the response to the ReceiveMessage request contains at least one of the available messages, up to the maximum number of messages specified in the ReceiveMessage action.
- Long polling returns messages as soon any message becomes available

Delay Queue/Delay Seconds

- Lets you postpone the delivery of new messages for the specified # seconds
- Message is i**nvisible to consumers** for the duration of the delay period
- can be from 0 to 900 seconds (i.e. 0 to 15 minutes)
- For Standard Q - is **not 'retroactive'**
  - does not affect the delay of messages already in Q
- For FIFI Q - it is 'retroactive'
  - does affect the messages which are already in Q
- It is ALSO POSSIBLE to set delay seconds at message level - Override)

Message Visibility

- Message is hidden only **after a message is consumed from the Q**
- To ensure that message is not processed more than once

Delay Q and Message Visibility - Difference ?

Parameter	Delay Q	Message Visibility
When it is hidden ?	Message is hidden when it is first added to Q	Message is hidden only after a message is consumed from the Q

Inflight messages

- Message is inflight after it is received from a Q by a consumer but NOT YET DELETED from the Q
- Messages which are getting processed by by consumer

Retention Period , Purging and Message Attributes

- Default is 4 days , min : 1 min , max : 14 days
- After retention period is over the message is no more available in Q
- Purging
  - delete all messages from Q
- Message Attributed
  - Allows to provide structure metadata
    - timestamp
    - geospatial data
    - custom data
  - Lets you decide how to handle the message without having to first process the message body
  - Upto 10 attributes/message

Dead Letter Q

- Queue that other queues can target to send messages that for some reason could not be successfully processed
- Provides Ability to sideline and isolate the unsuccessfully processed messages
- Multiple Source Queue can have single Dead Letter Queue

Programming Interface

Using Q

- Create Q
- Send Message to Q
- Receive Q

Limits

SQS Limits

Parameter	Value	Remarks
Message text	256 KB	
Inflight messages *Messages are inflight after they have been received from the queue by a consuming component, but have not yet been deleted from the queue*	120,000	If you reach the limit - OverLimit error message Remedy : ● delete messages from the queue after they have been processed. ● Increase the number of queues to process the messages.
Delay Queue - `DelaySeconds`	0 to 900 seconds (15 minutes)	

Pricing

SQS Pricing Characteristics

- **Requests Based Pricing**
  - 1 million Amazon SQS requests for free each month

	Price per 1 Million Requests after Free Tier (Monthly)
Standard Queue	$0.40 ($0.00000040 per request)
FIFO Queue	$0.50 ($0.00000050 per request)

- **Data Transfer**
  - Data transfer *in* and *out* refers to transfer into and out of Amazon SQS.
  - Data transferred between Amazon SQS and Amazon EC2 within a single region is free of charge (that is, $0.00 per GB).
  - Data transferred between Amazon SQS and Amazon EC2 in different regions is charged at Internet Data Transfer rates on both sides of the transfer

For further details reer https://aws.amazon.com/sqs/pricing/

Thank You