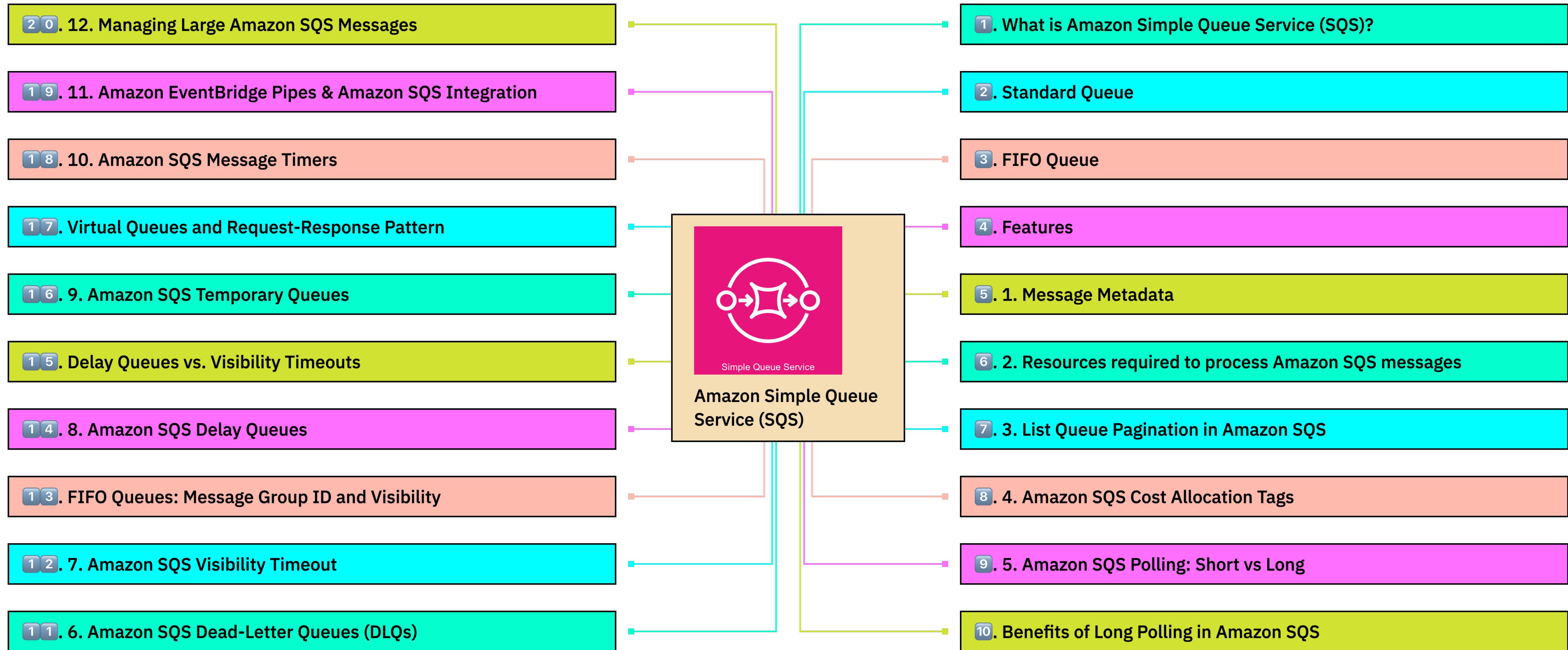
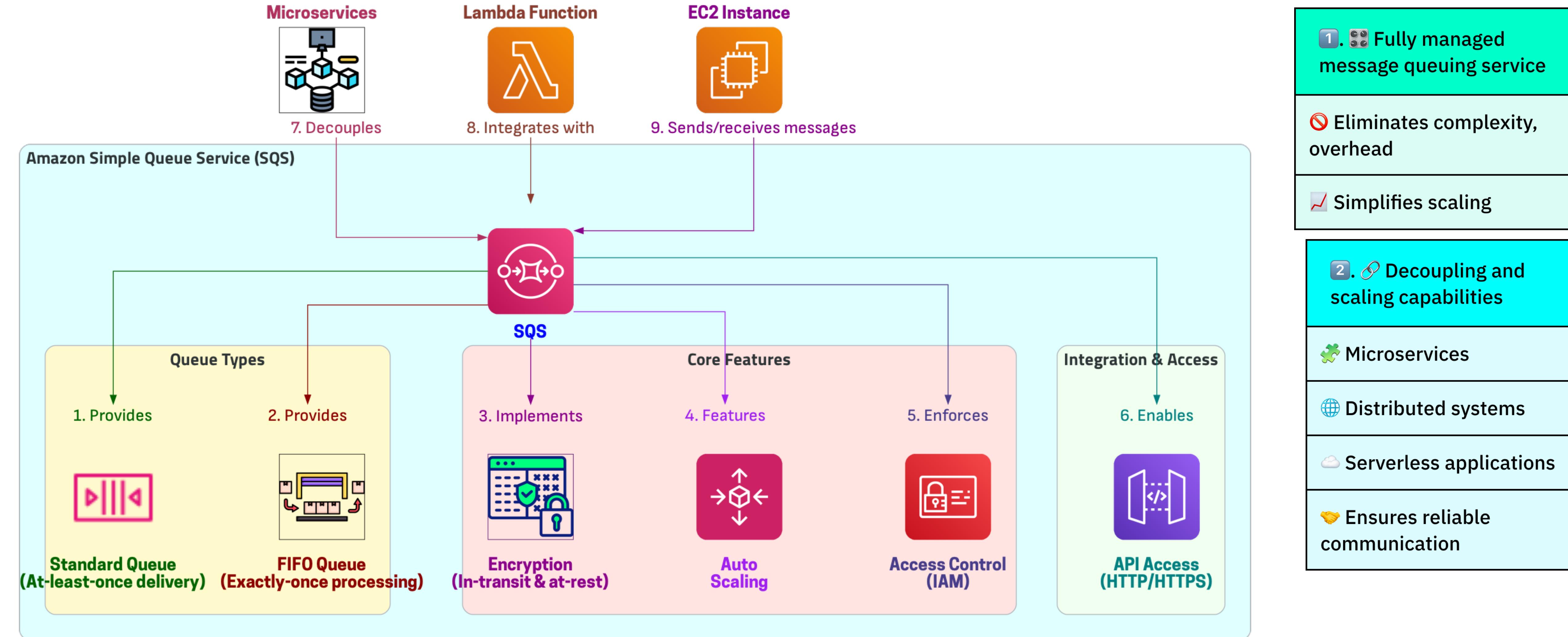


Amazon Simple Queue Service (SQS)

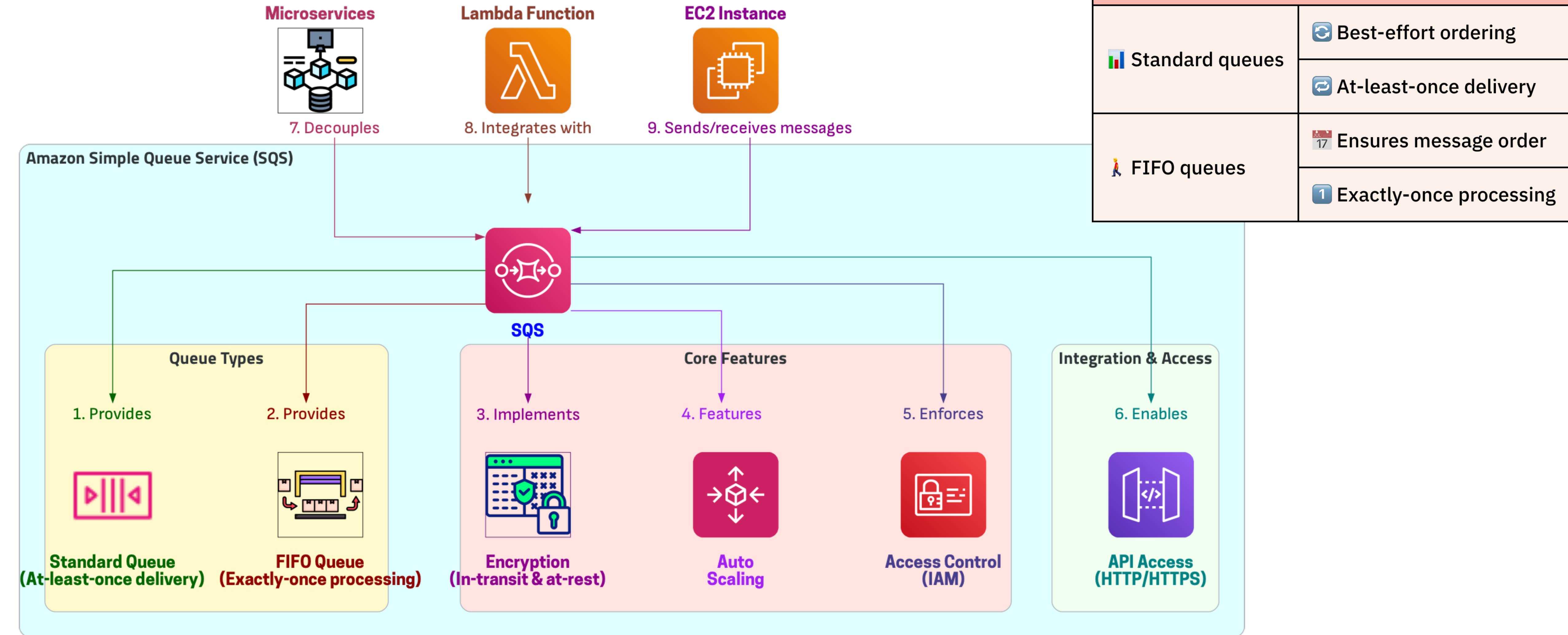
Table of Contents



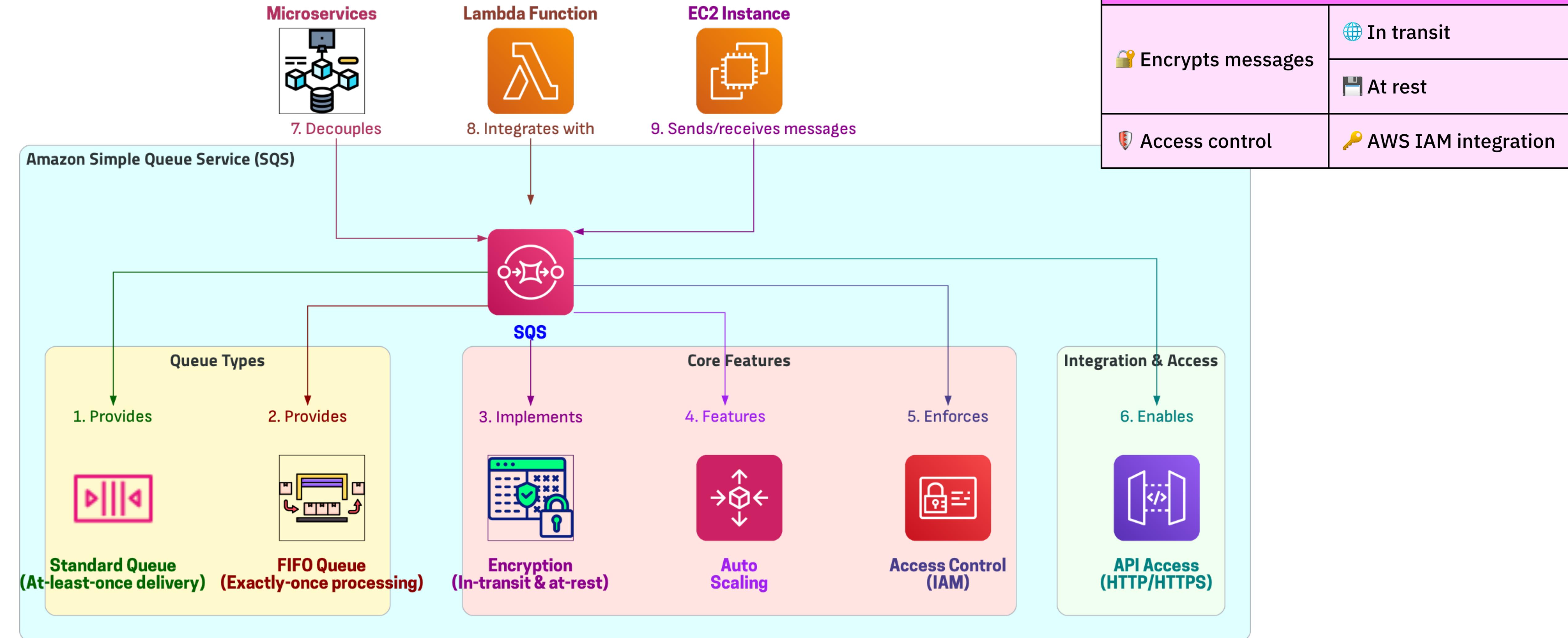
What is Amazon Simple Queue Service (SQS)?



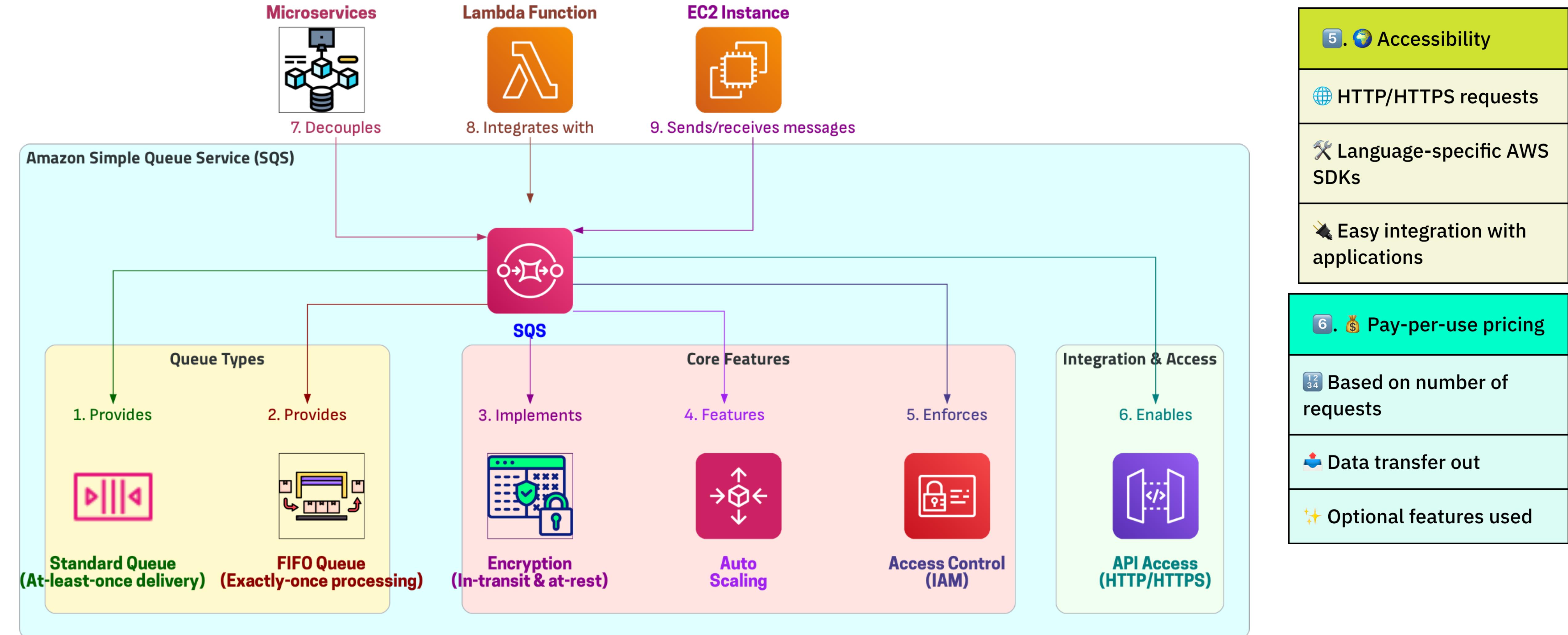
What is Amazon Simple Queue Service (SQS)?



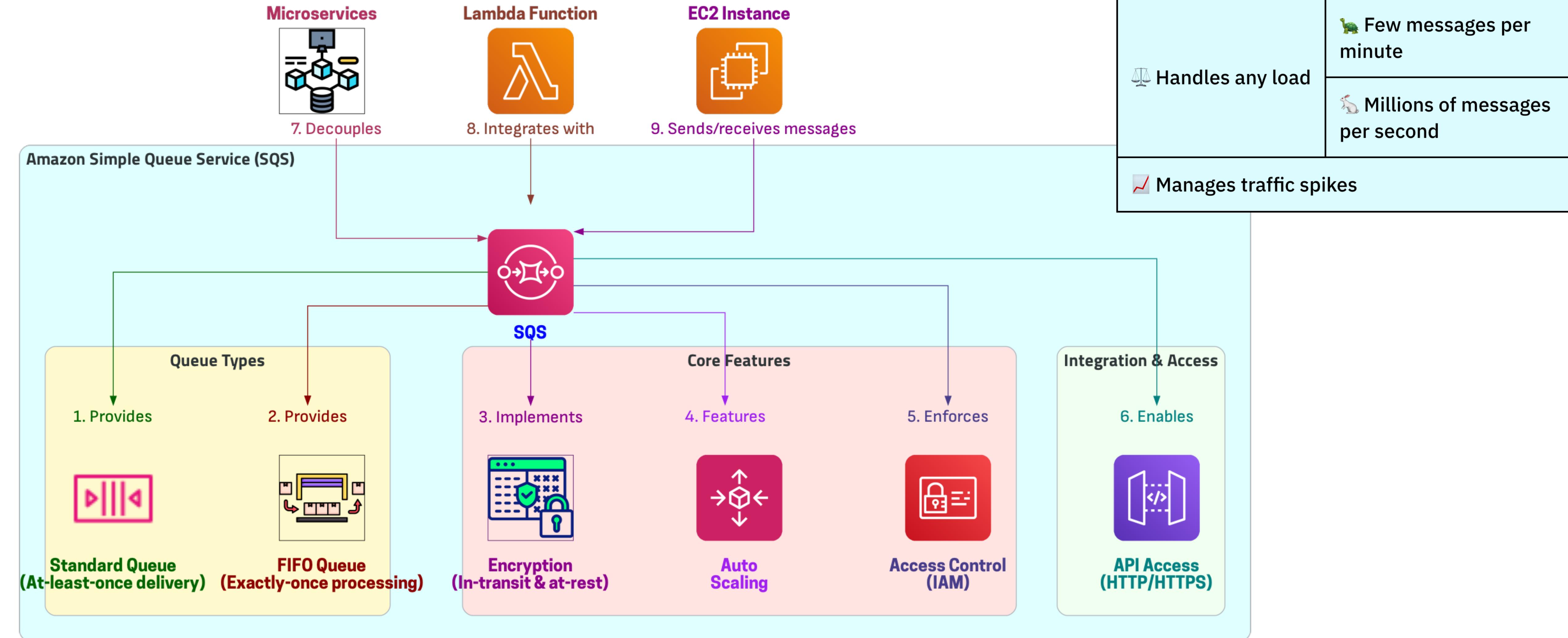
What is Amazon Simple Queue Service (SQS)?



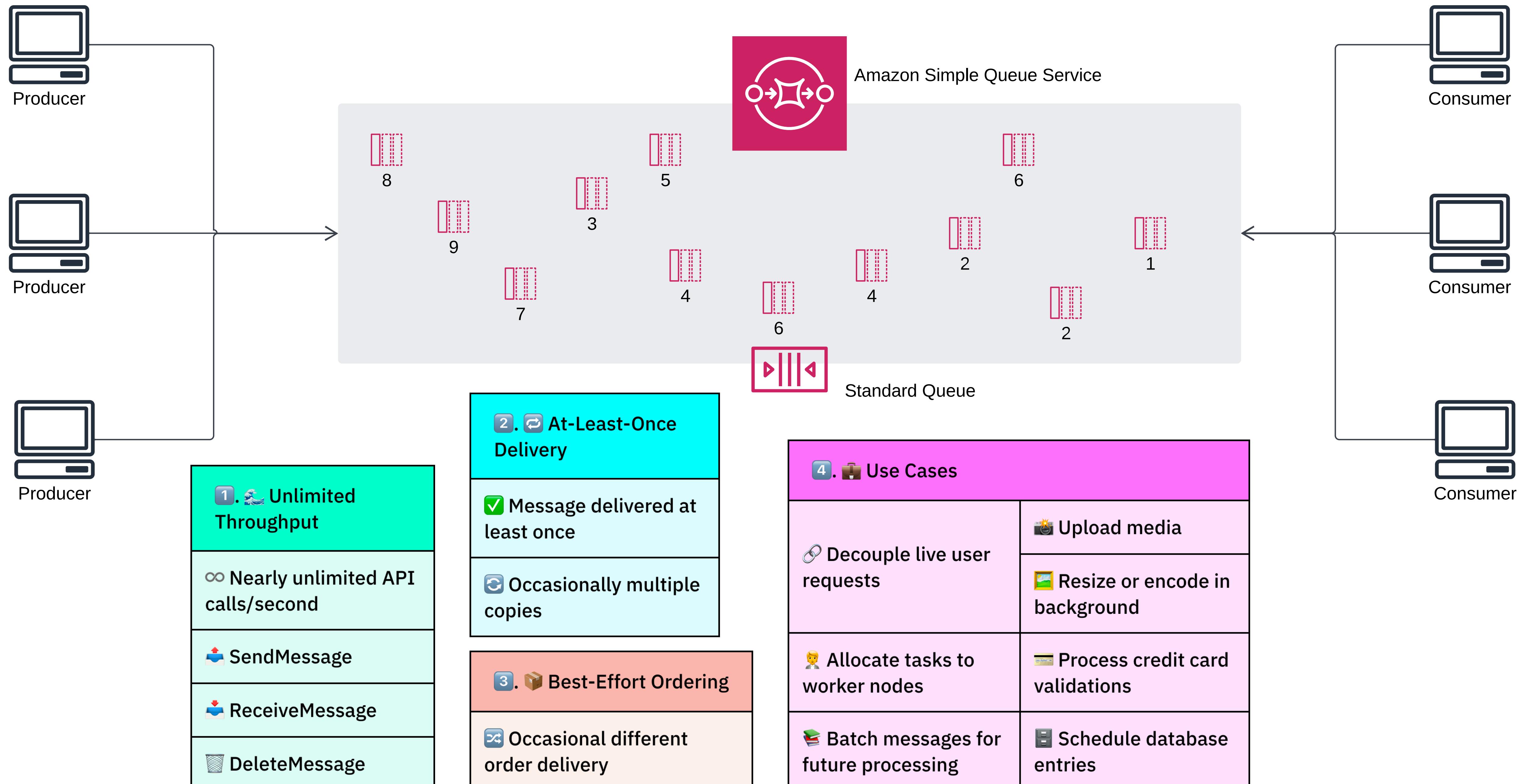
What is Amazon Simple Queue Service (SQS)?



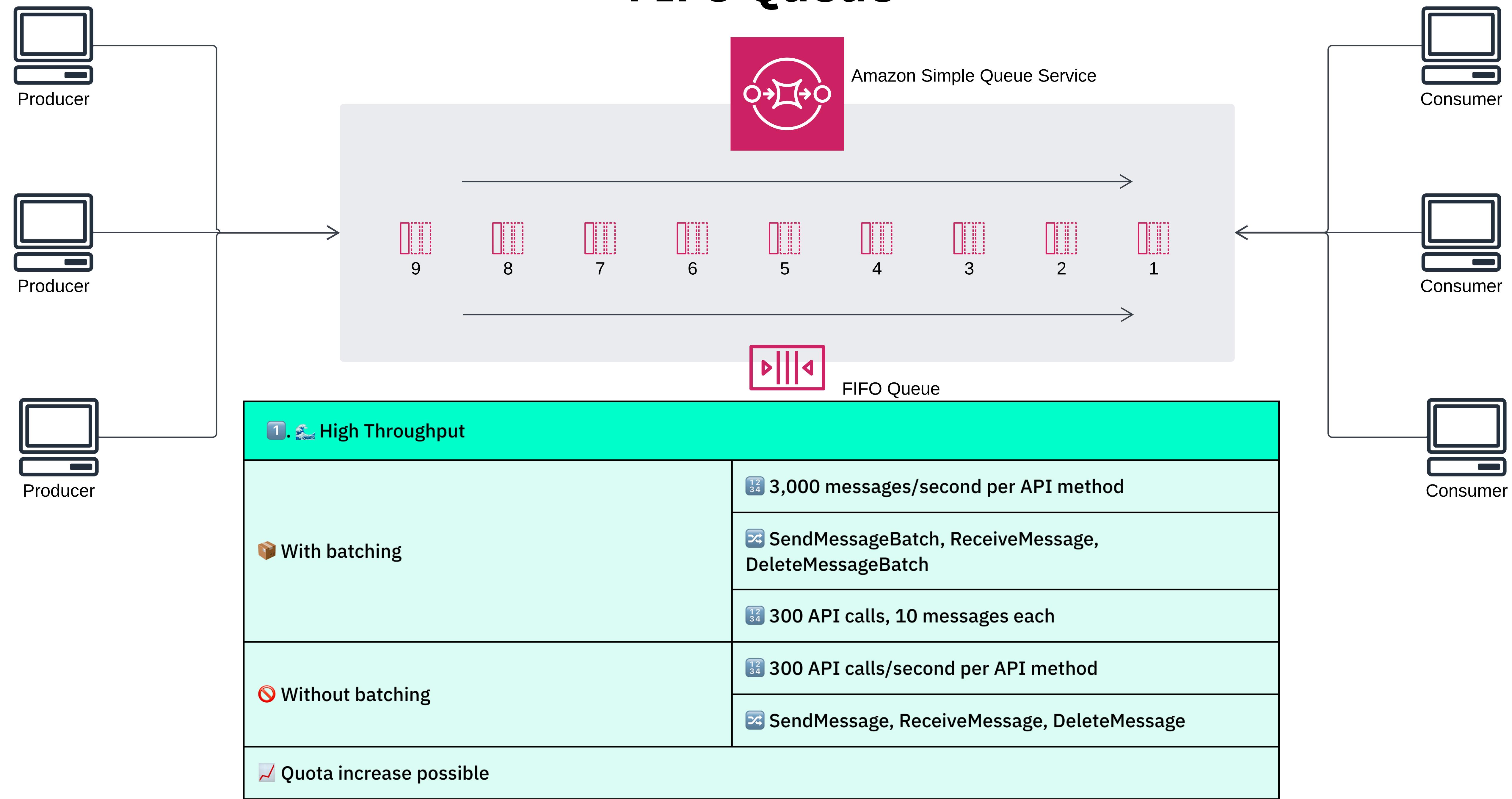
What is Amazon Simple Queue Service (SQS)?



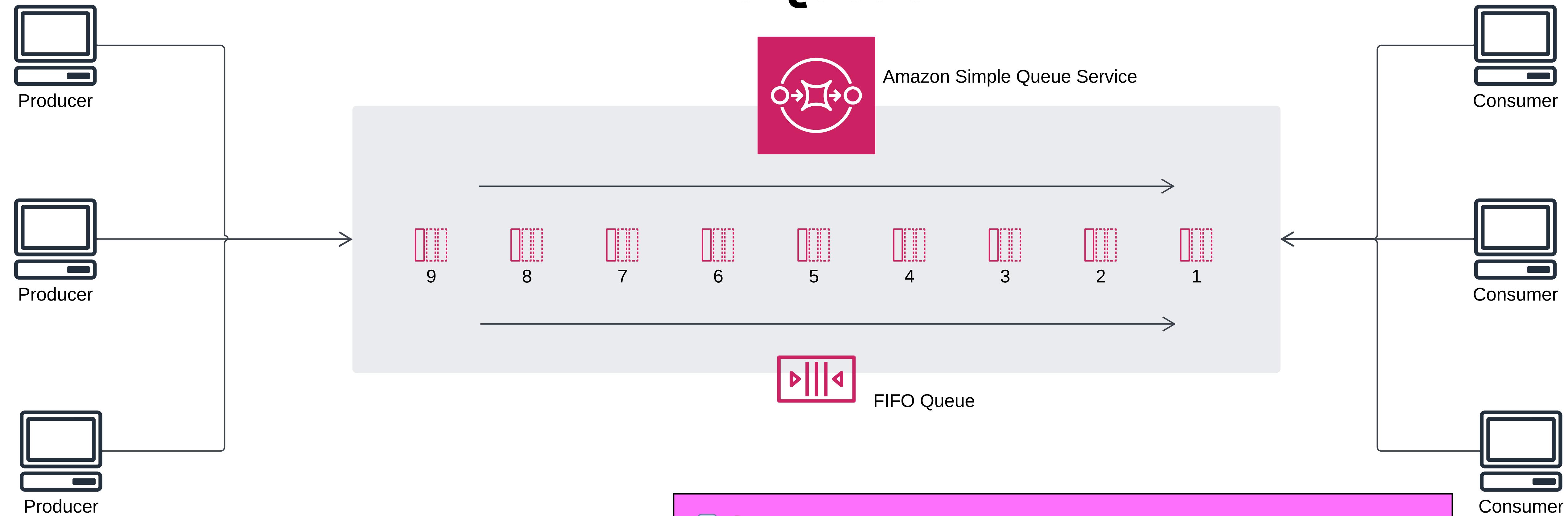
Standard Queue



FIFO Queue



FIFO Queue



2. ⚡ Exactly-Once Processing
✓ Message delivered once
⟳ Available until processed and deleted
🚫 No duplicates introduced

3. 📦 First-In-First-Out Delivery
🔒 Strict order preservation
➡️ Sending order maintained
⬇️ Receiving order maintained

4. 💼 Use Cases	
🎮 User-entered commands	1234 Ensure correct execution order
💰 Product price updates	📊 Display correct prices
🎓 Student enrollment	1234 Apply modifications in order
	🔒 Prevent premature course enrollment
	📝 Ensure account registration first

Features - 1

1 Message metadata

2 Resources required to process messages

3 List queue pagination

4 Cost allocation tags

5 Short and long polling

6 Dead-letter queues

7 Visibility timeout

8 Delay queues

9 Temporary queues

10 Message timers

Features - 2

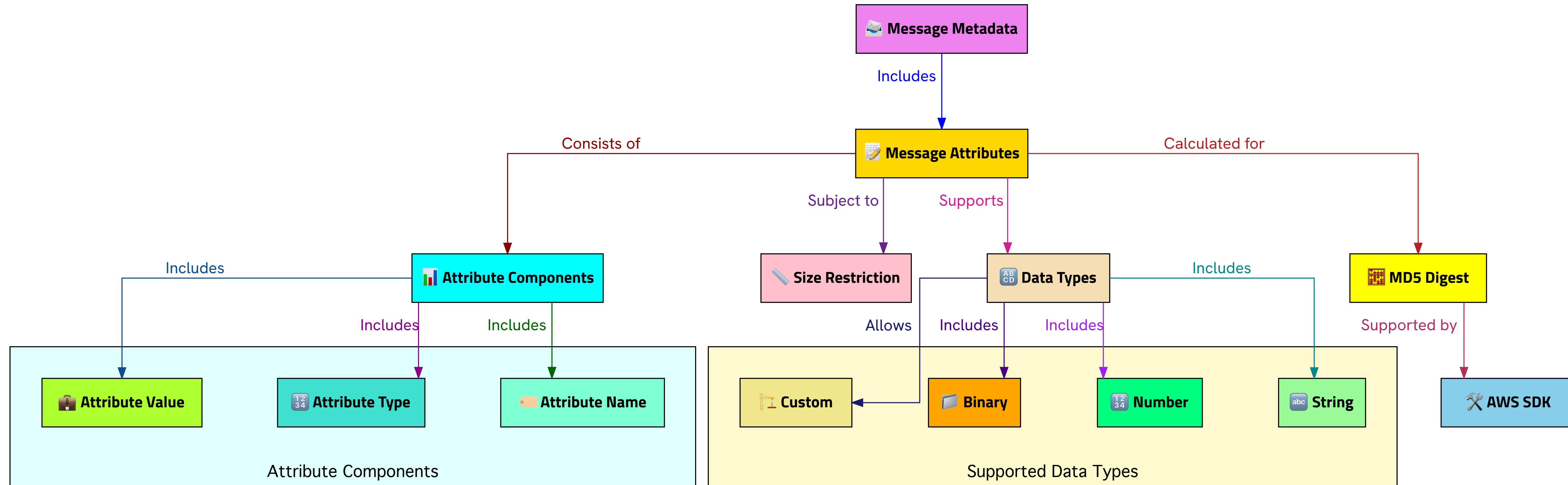
11

Accessing EventBridge pipes

12

Managing large messages

1. Message Metadata



1. Message Attributes

Up to 10 attributes per message

Structured metadata	Timestamps
	Geospatial data
	Signatures
	Identifiers

Attribute Properties
Optional and separate from message body
Sent alongside message body
Included in 256 KB size limit

3. Attribute Components

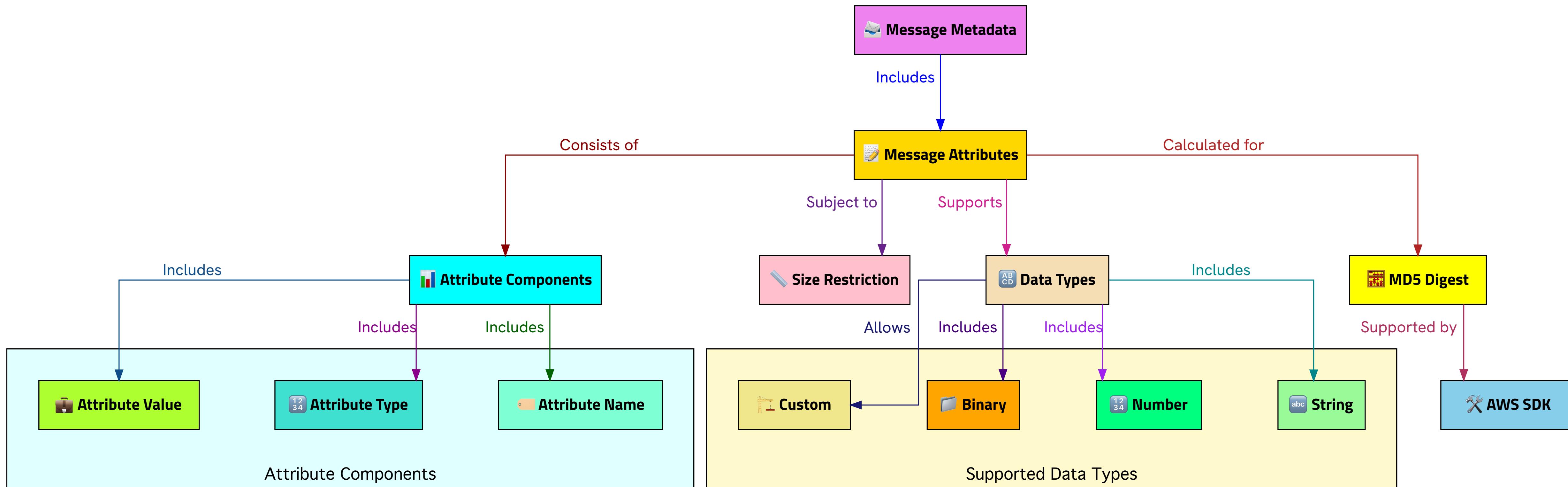
Name (up to 256 characters)

Type (String, Number, Binary)

Value

No empty or null components allowed

1. Message Metadata



4. Attribute Name Rules

- Allowed characters: A-Z, a-z, 0-9, _, -, .
- No AWS. or Amazon. prefix
- Case-sensitive
- Must be unique

5. Supported Data Types

- String (Unicode text)
- Number (38-digit precision)
- Binary (any binary data)

6. Custom Data Types

- Created by appending label to data type
- Examples: Number.int, Binary.png
- Not interpreted or validated by SQS

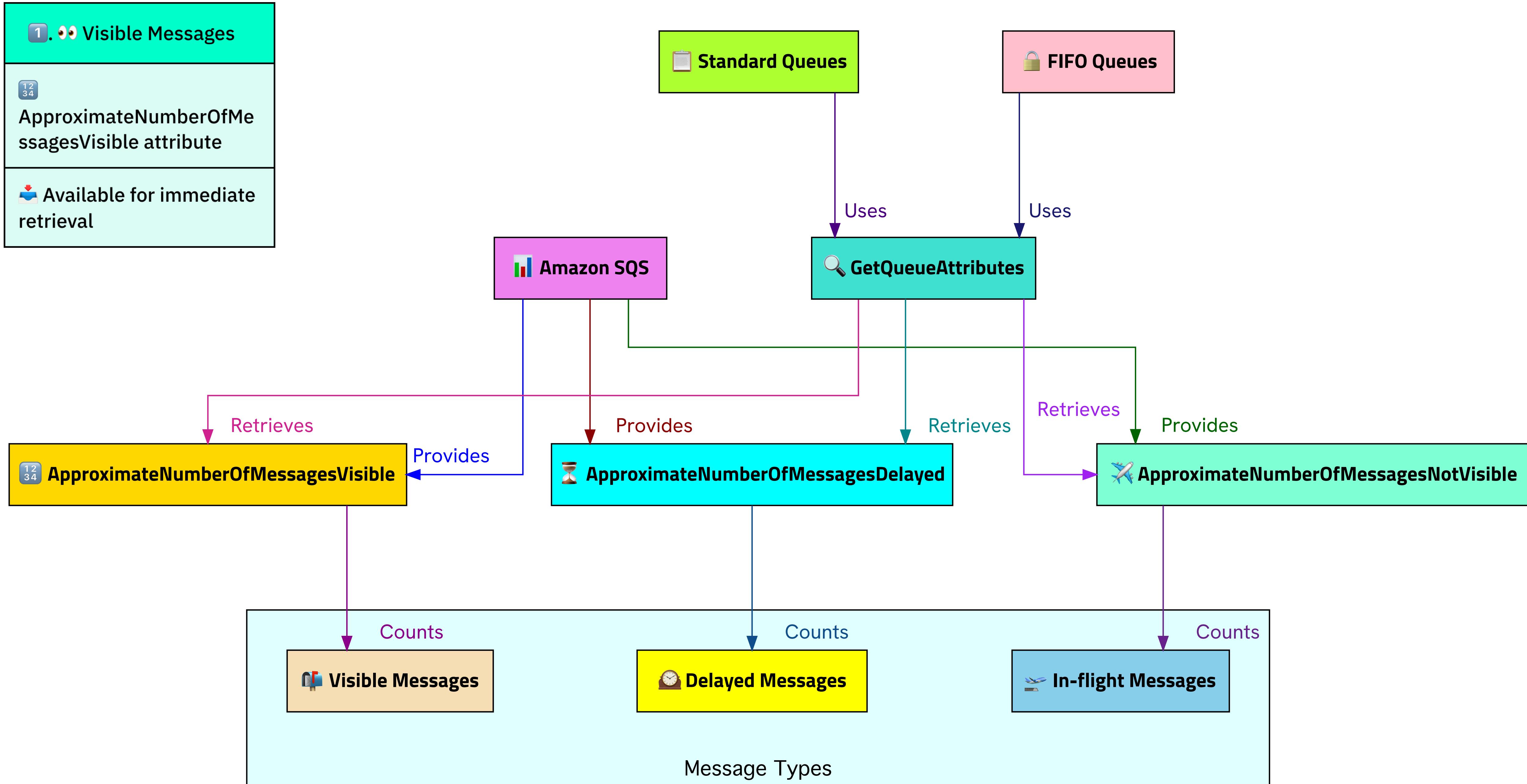
7. MD5 Digest Calculation

- Sort attributes by name
- Encode Name, Type, Value into buffer
- Compute digest of entire buffer

8. SDK Support

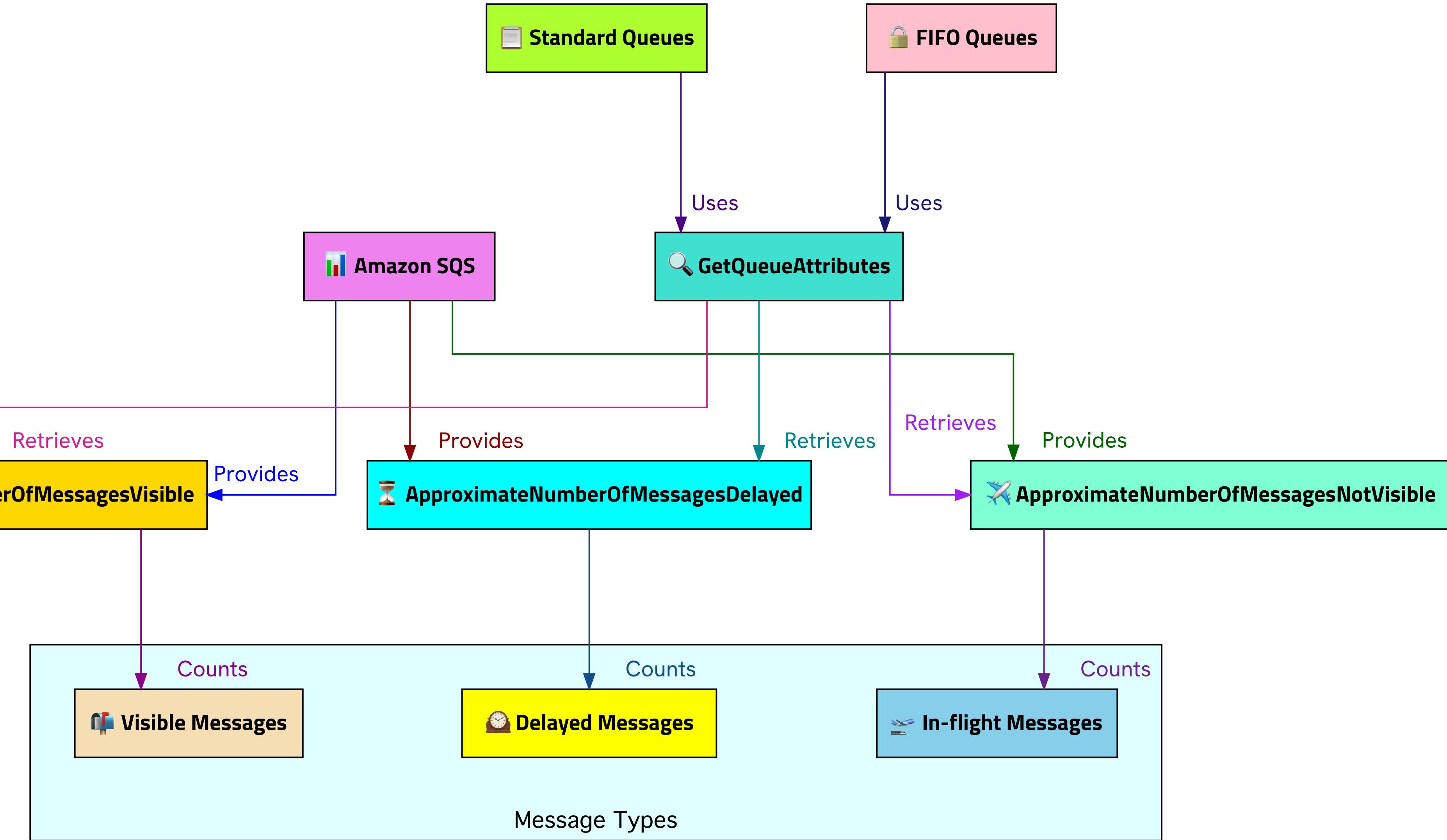
Java SDK: MessageMD5ChecksumHandler class
Other SDKs: Manual calculation required
Query API: Manual calculation required

2. Resources required to process Amazon SQS messages

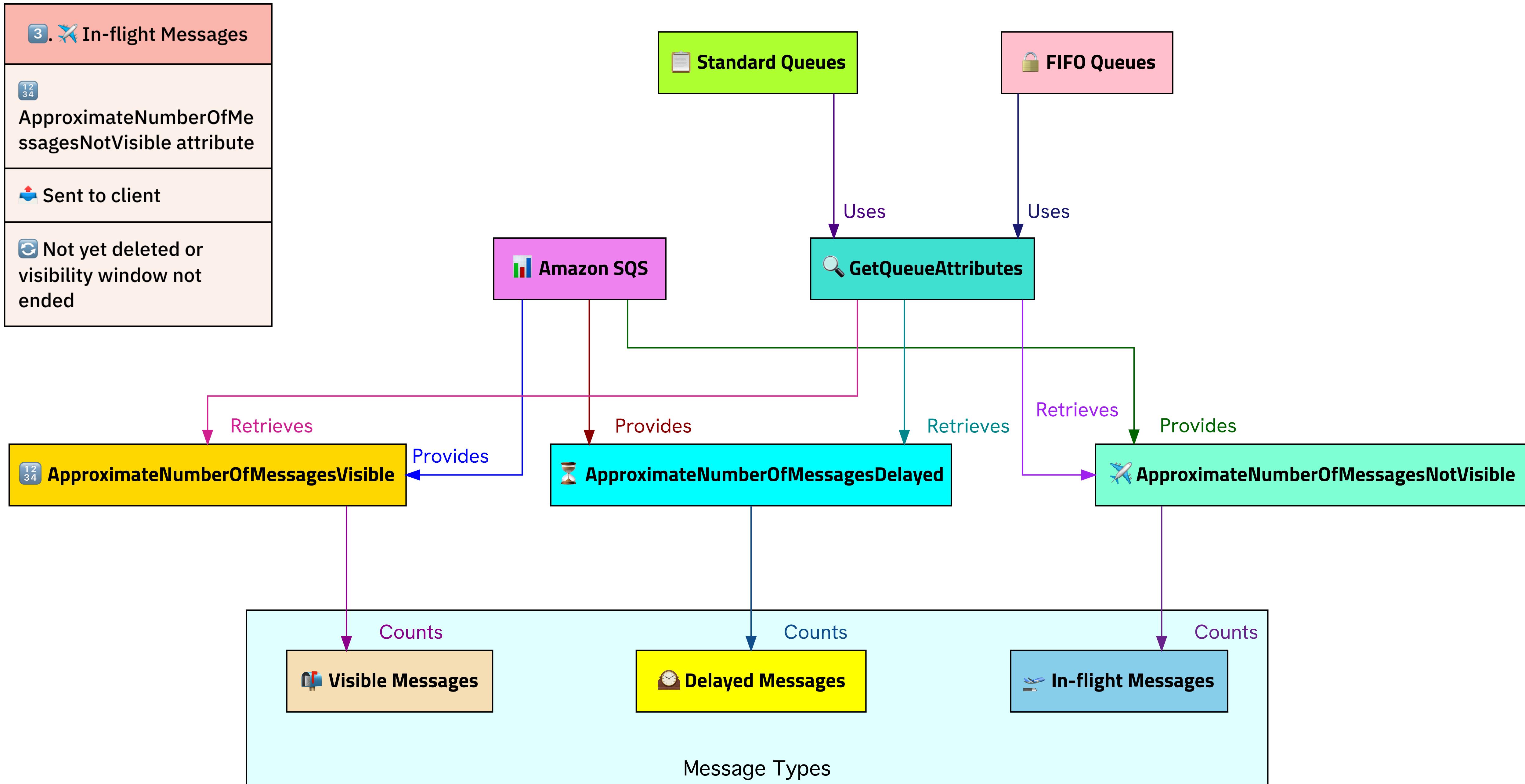


2. Resources required to process Amazon SQS messages

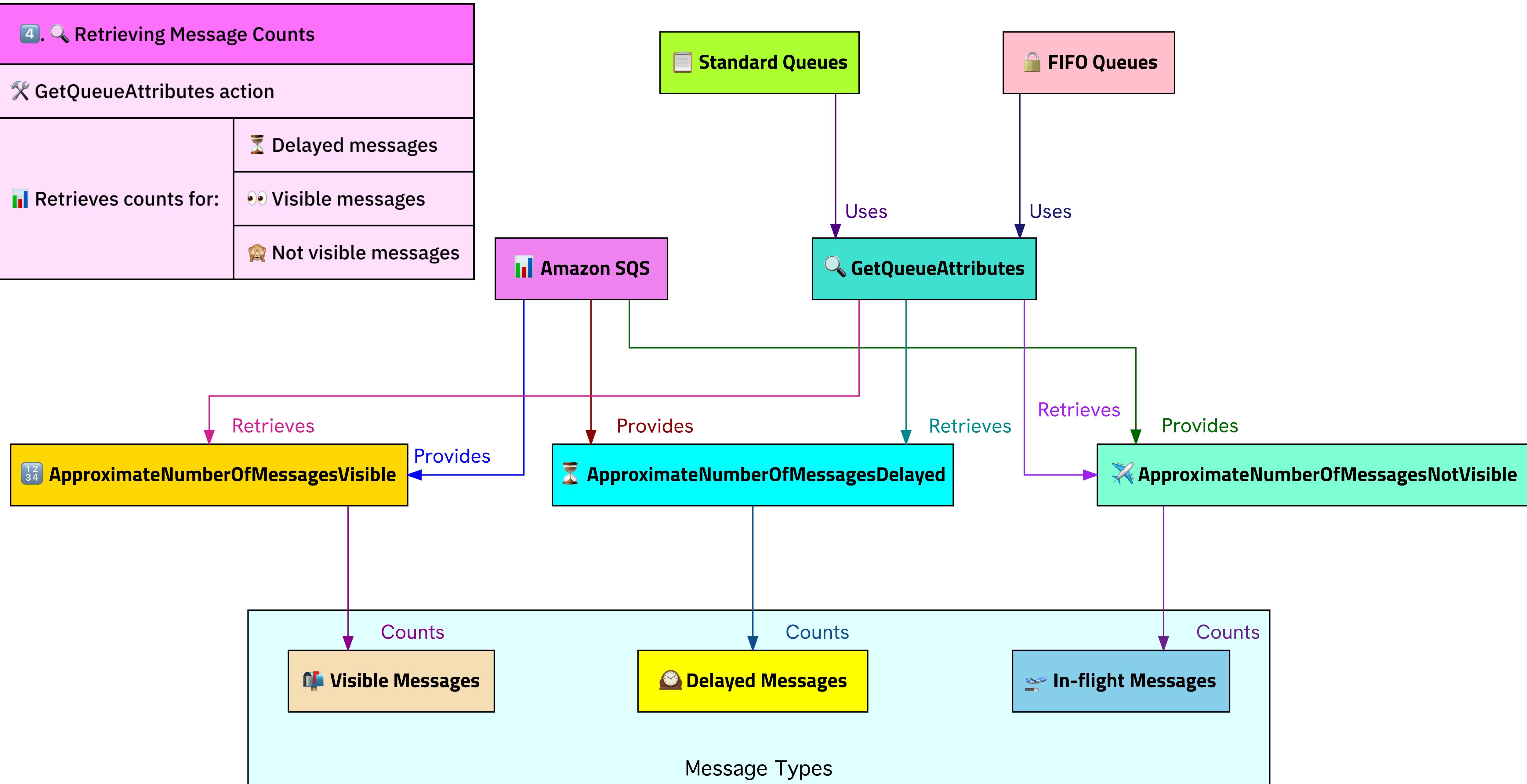
2. ⏳ Delayed Messages	
12 34	ApproximateNumberOfMessagesDelayed attribute
⌚	Not available for immediate reading
Occurs in:	<ul style="list-style-type: none">⌚ Delay queue configuration✉️ Message sent with delay parameter



2. Resources required to process Amazon SQS messages

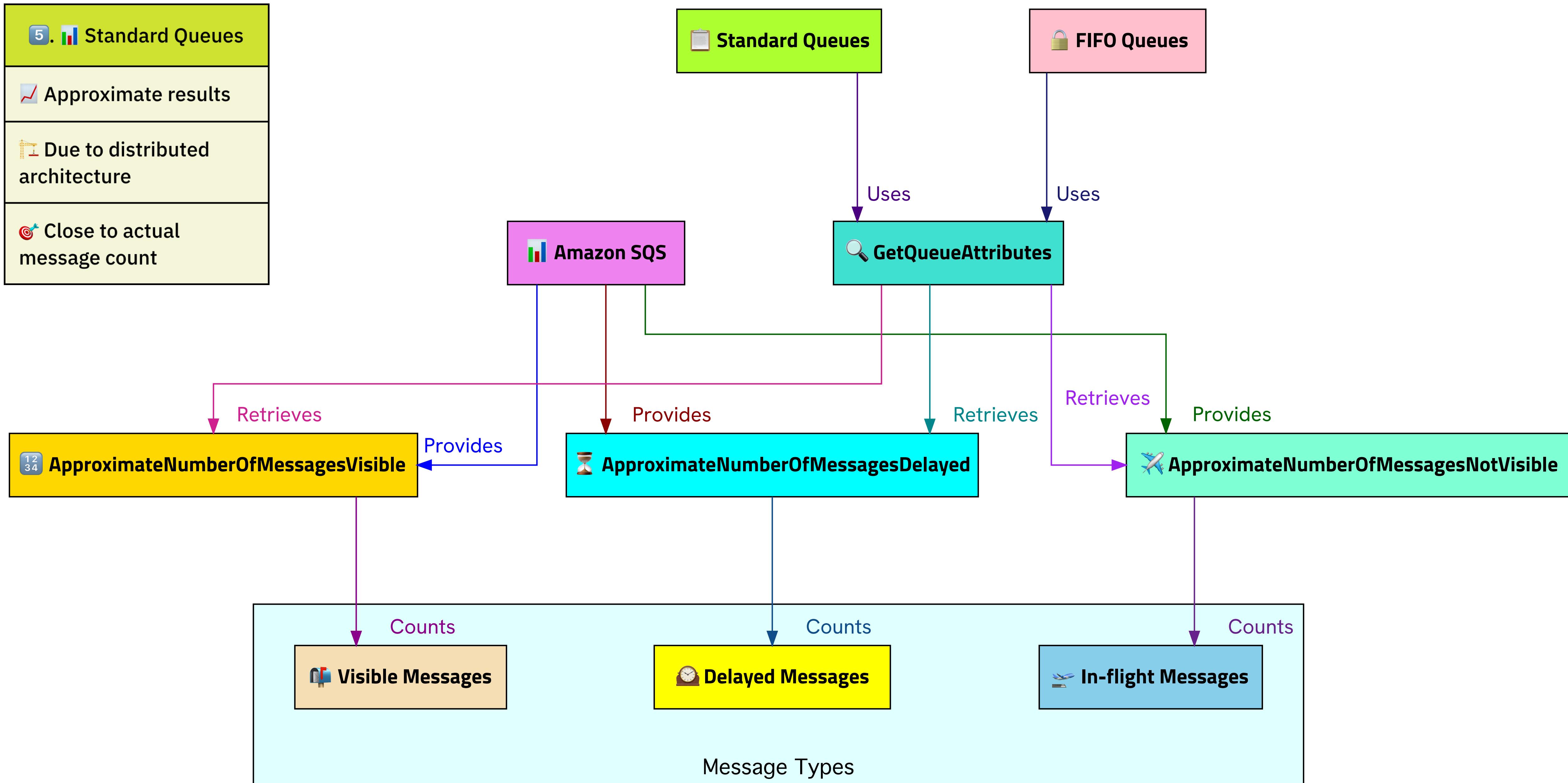


2. Resources required to process Amazon SQS messages

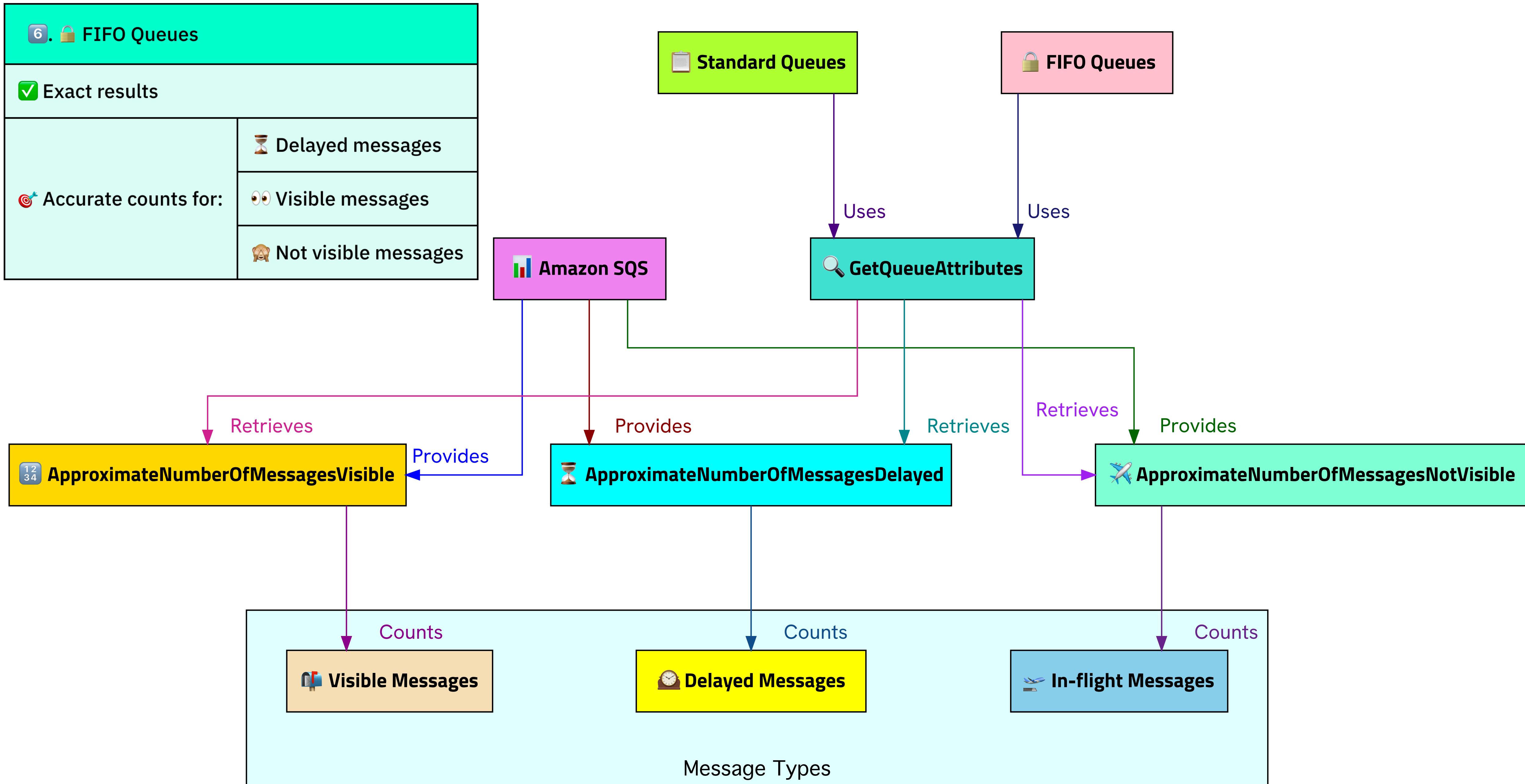


2. Resources required to process Amazon SQS messages

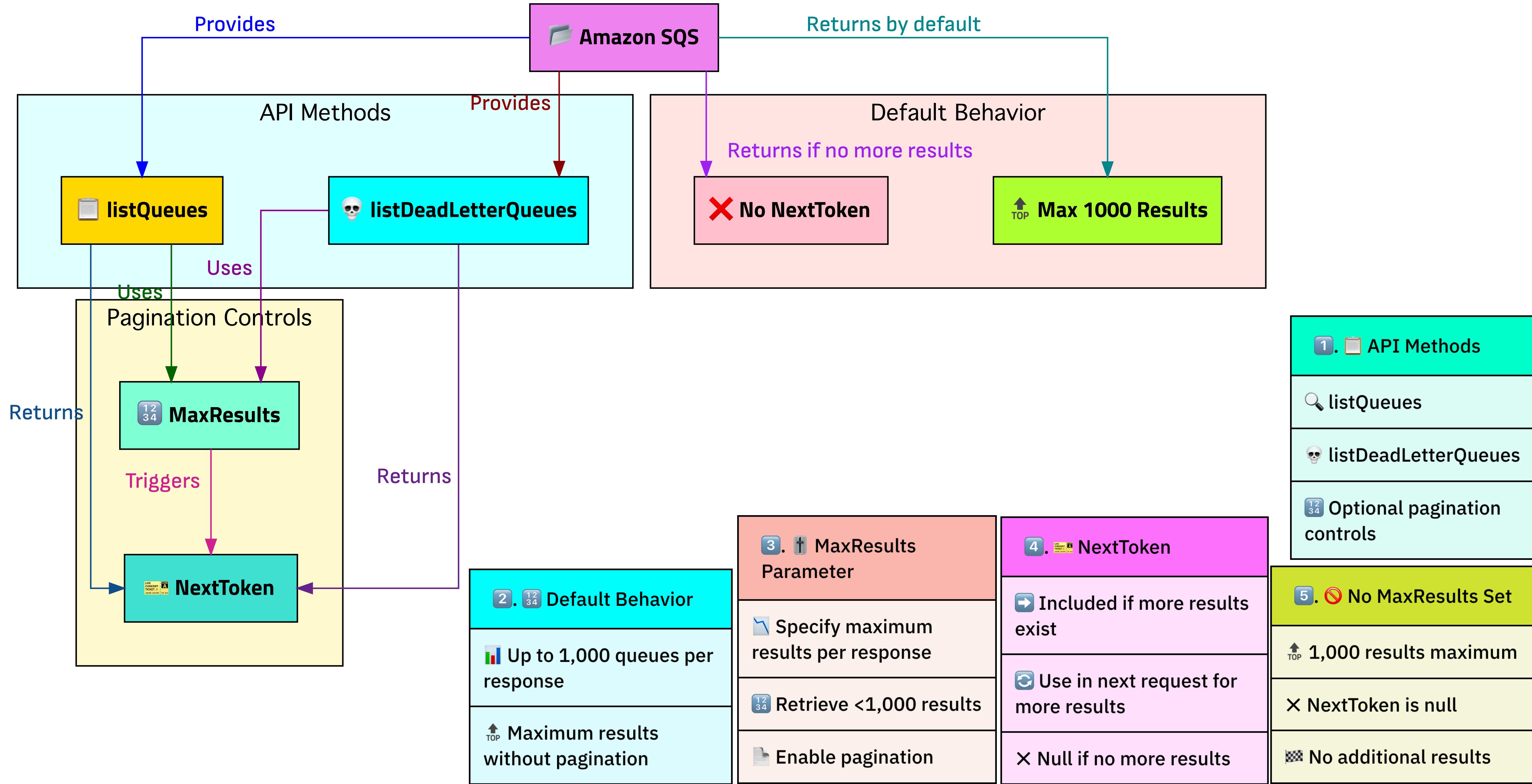
5. Standard Queues
Approximate results
Due to distributed architecture
Close to actual message count



2. Resources required to process Amazon SQS messages



3. List Queue Pagination in Amazon SQS



4. Amazon SQS Cost Allocation Tags

1. Organization and Identification

Cost allocation purposes

Manage multiple queues

2. Metadata Tags

Identify queue purpose

Identify queue owner

Identify queue environment

Easier categorization and tracking

3. Amazon SQS Console

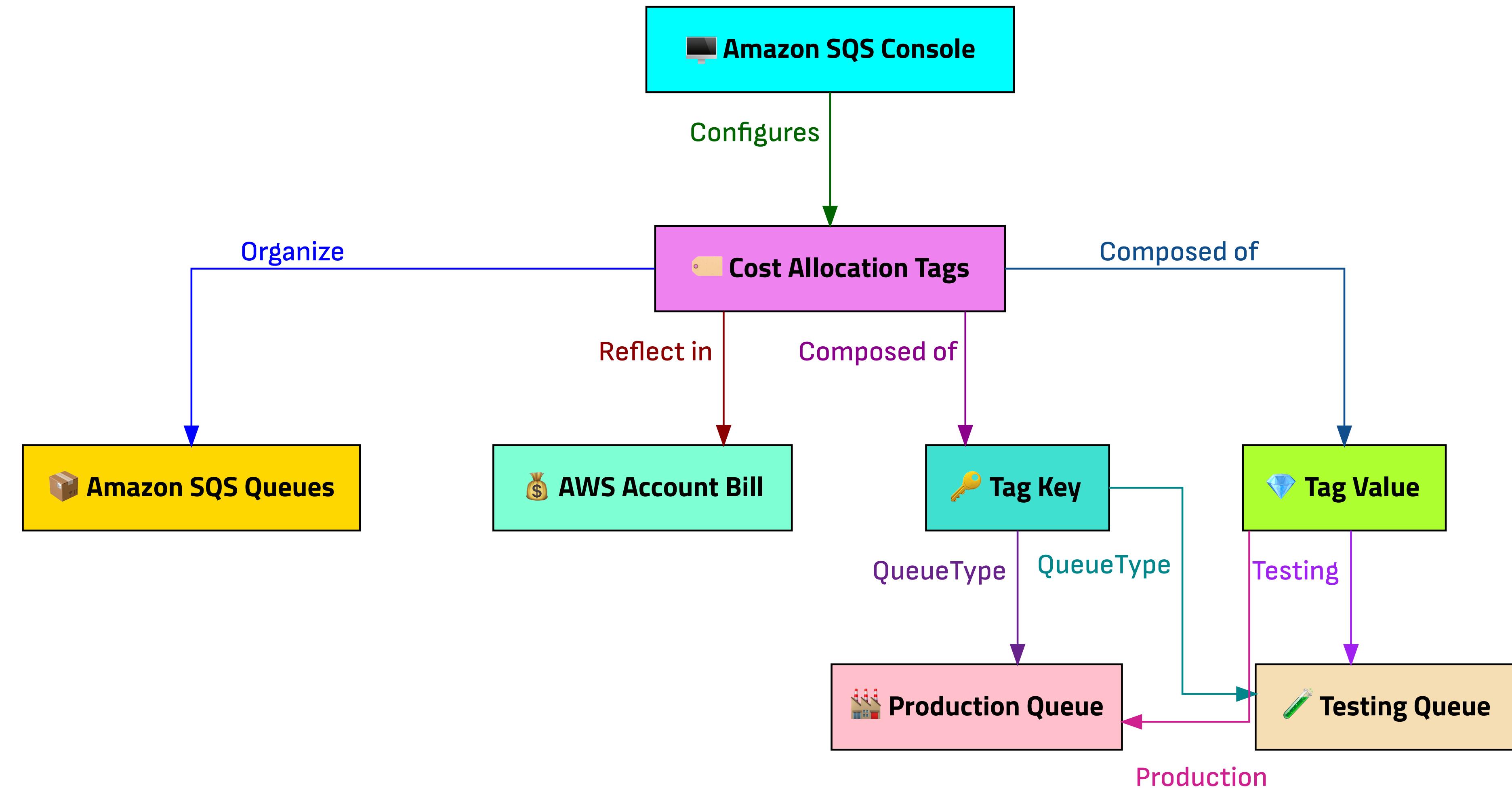
User-friendly interface

Configure cost allocation tags

+ Add tags

Modify tags

X Remove tags

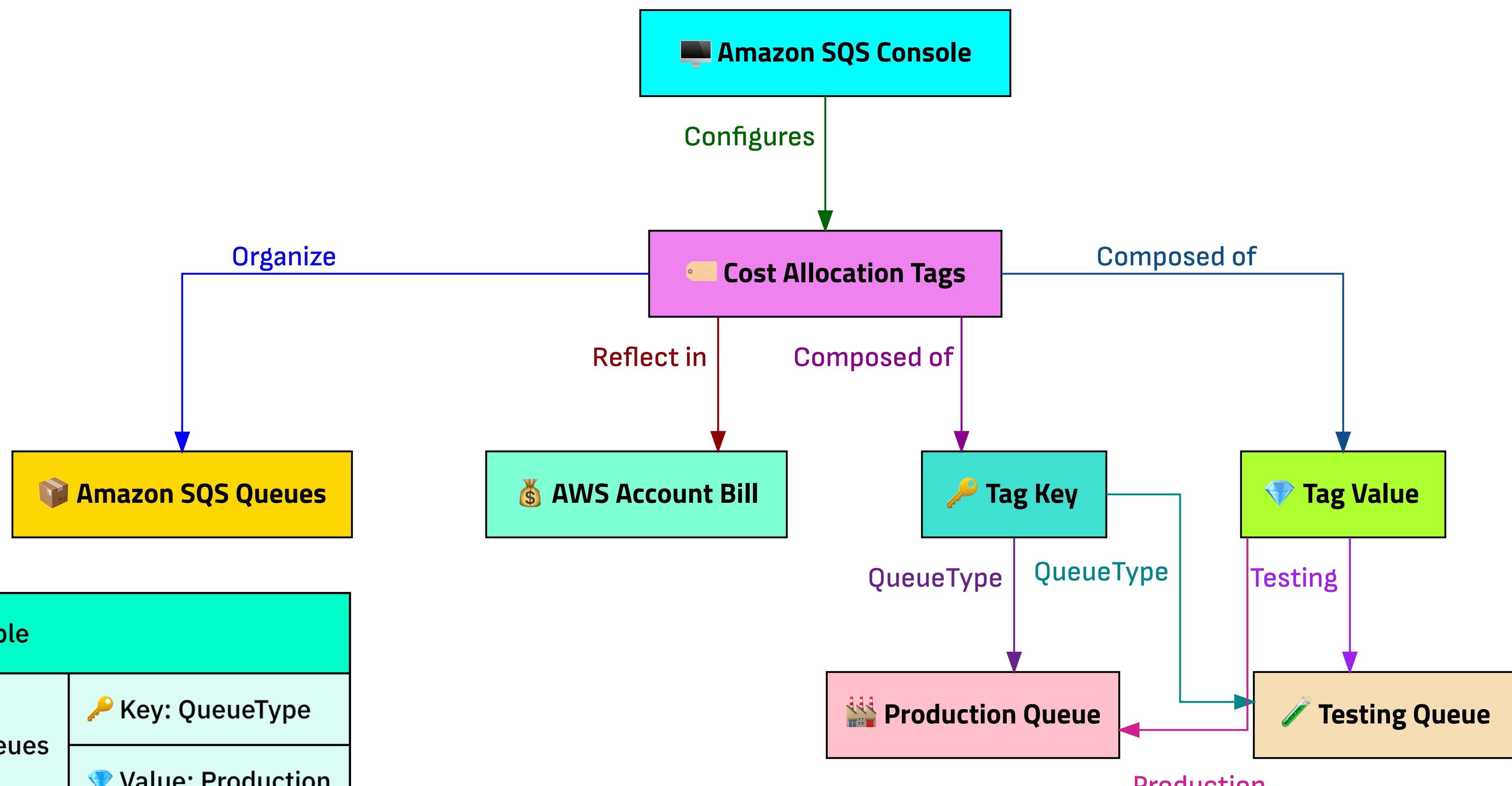


4. Amazon SQS Cost Allocation Tags

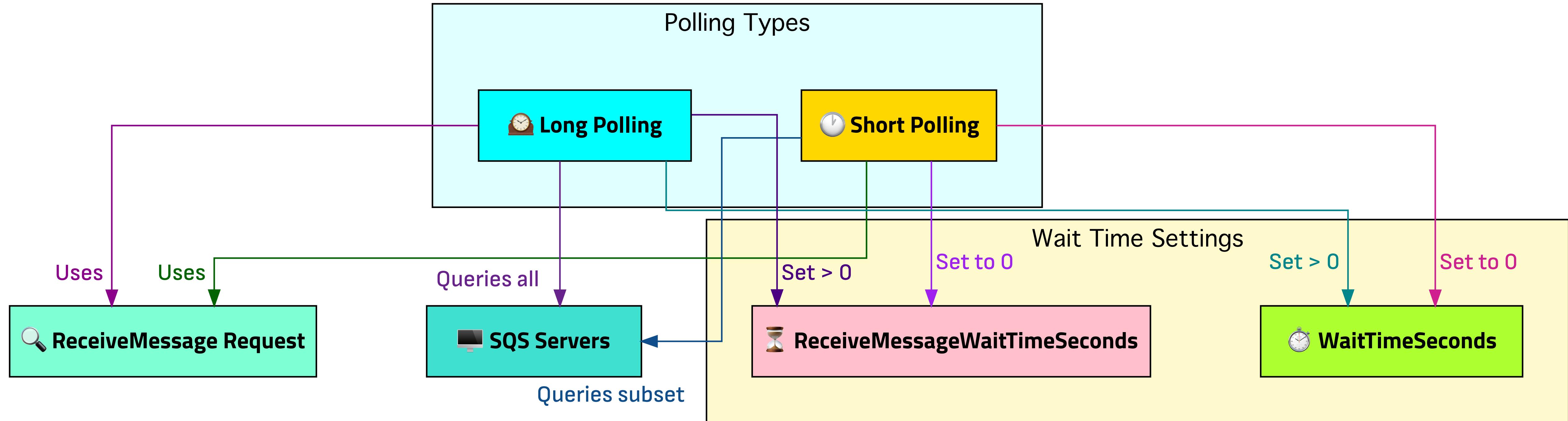
4. 💰 AWS Account Bill
Include tag keys and values
Organize bill to reflect cost structure
Align with tagged queues

6. 🌟 Tag Example
Production queues
Key: QueueType Value: Production
Testing queues
Key: QueueType Value: Testing
Easy differentiation between environments

5. 🔑 Tag Structure
Key-value pairs
Key: Tag category
Value: Specific tag within category



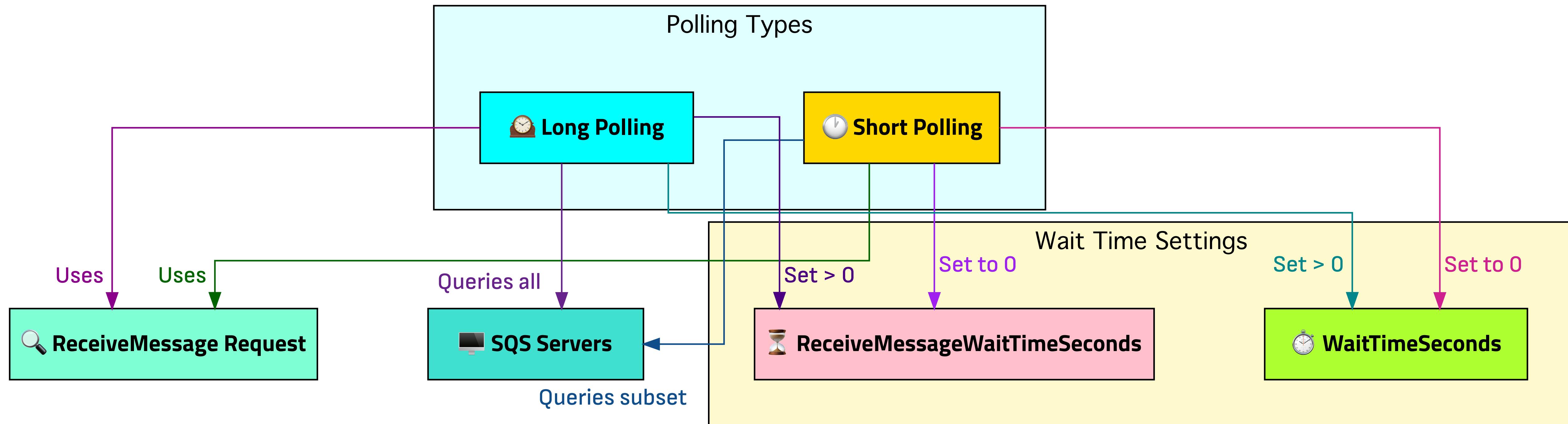
5. Amazon SQS Polling: Short vs Long



1. Short Polling	
 Queries subset of servers	 Based on weighted random distribution
 Immediate response	 Even if no messages found
 Subsequent request	 For <1000 messages in queue  Will return messages

2. ⏱️ Long Polling	
 Queries all servers	 Collects at least one message
 Waits for messages	 Up to max specified in request
 Empty response	 Only if polling wait time expires

5. Amazon SQS Polling: Short vs Long



3. Configuration		
⌚ WaitTimeSeconds parameter	0 Set to 0 for short polling	📞 Directly in ReceiveMessage call 🔧 Or via ReceiveMessageWaitTimeSeconds queue attribute
+ Greater than 0 for long polling		1234 Maximum of 20 seconds



Benefits of Long Polling in Amazon SQS

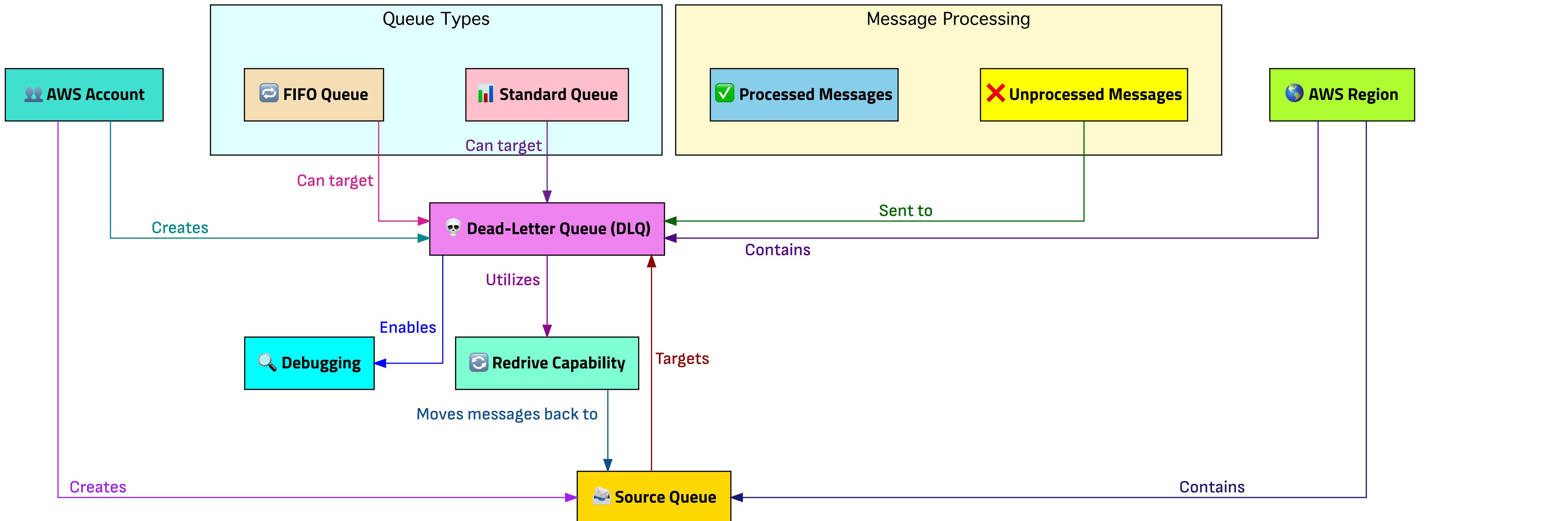
1. ⏳ Wait for message availability	
<p>⌚ Waits until message is available</p>	
<p>✉️ Returns at least one message</p>	<p>↑ Up to specified maximum</p>
<p>🚫 Minimizes empty responses</p>	

2. 🌐 Queries all servers	
<p>⚡ Reduces false empty responses</p>	
<p>✅ Ensures message inclusion</p>	<p>🔍 Even if not on initial servers</p>

3. ⚡ Minimizes latency	
<p>🏃 Quick message retrieval</p>	
<p>⌚ Reduces time between</p>	<p>+ Message addition 👤 Consumer retrieval</p>

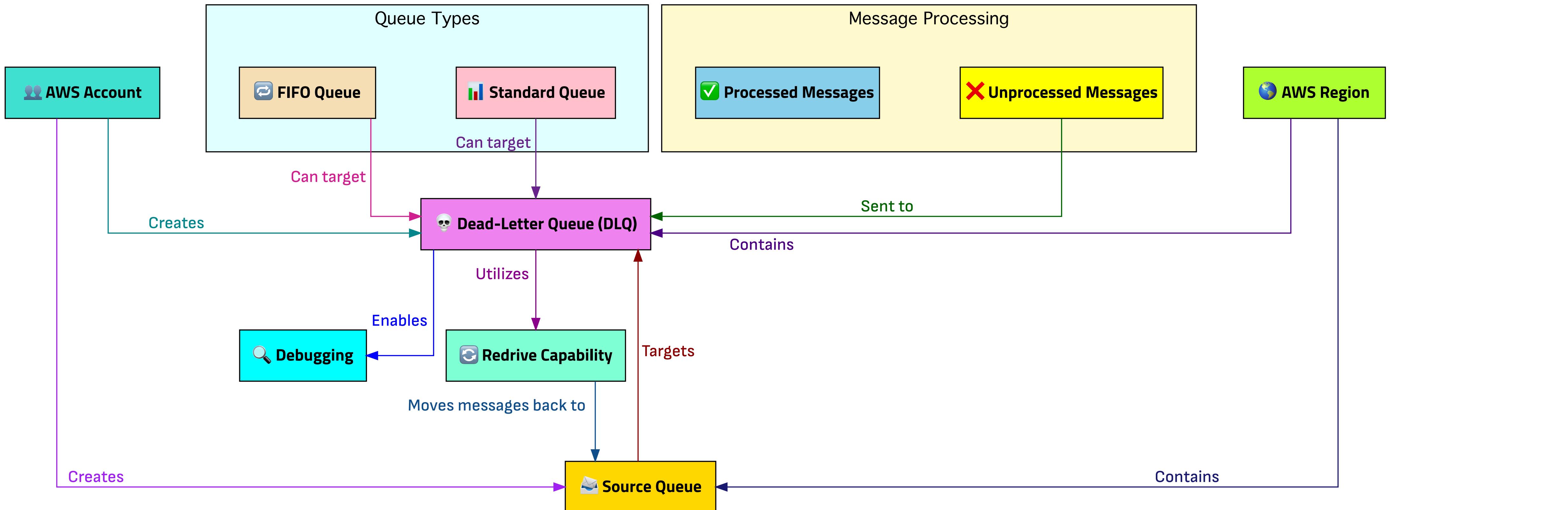
4. 💰 Cost reduction	
<p>🚫 Eliminates empty responses</p>	
<p>⚡ Reduces ReceiveMessage requests</p>	<p>✉️ More likely to return messages</p>

6. Amazon SQS Dead-Letter Queues (DLQs)



- | | | | |
|-----------------------------------|-------------------------------|---------------------------------|--------------------------------|
| 1. Purpose of DLQs | 2. Debugging benefits | 3. Investigation process | 4. DLQ creation |
| Target for unprocessed messages | Isolate unconsumed messages | Separate from main queue | Not automatic |
| Isolation of problematic messages | Determine processing failures | Analyze processing failures | Manual queue creation required |
| Debugging of issues | | Unexpected state changes | |

6. Amazon SQS Dead-Letter Queues (DLQs)

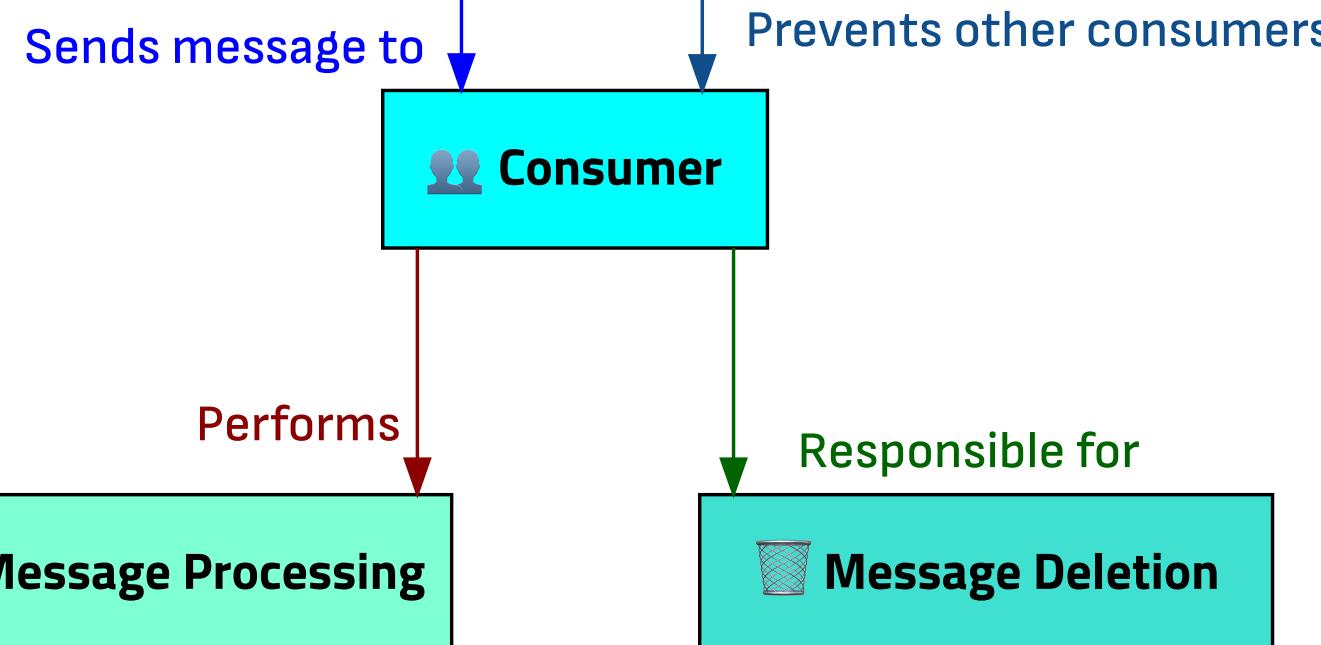
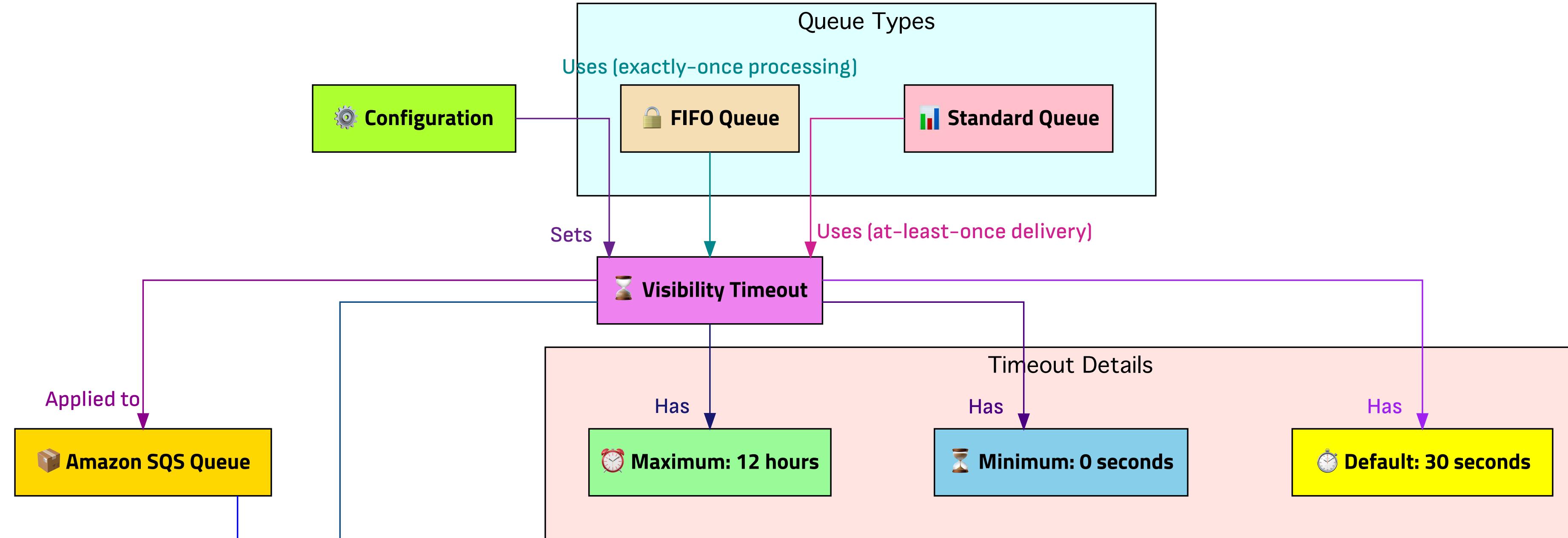


- 5. **Redrive capability**
- After debugging
- Move messages back to source queue**

6. Multiple queue support	
	Standard queues
	FIFO queues
	Single DLQ target
	Centralized handling

	For DLQ
	For source queues
	Ensure functionality
	Maintain security

7. Amazon SQS Visibility Timeout

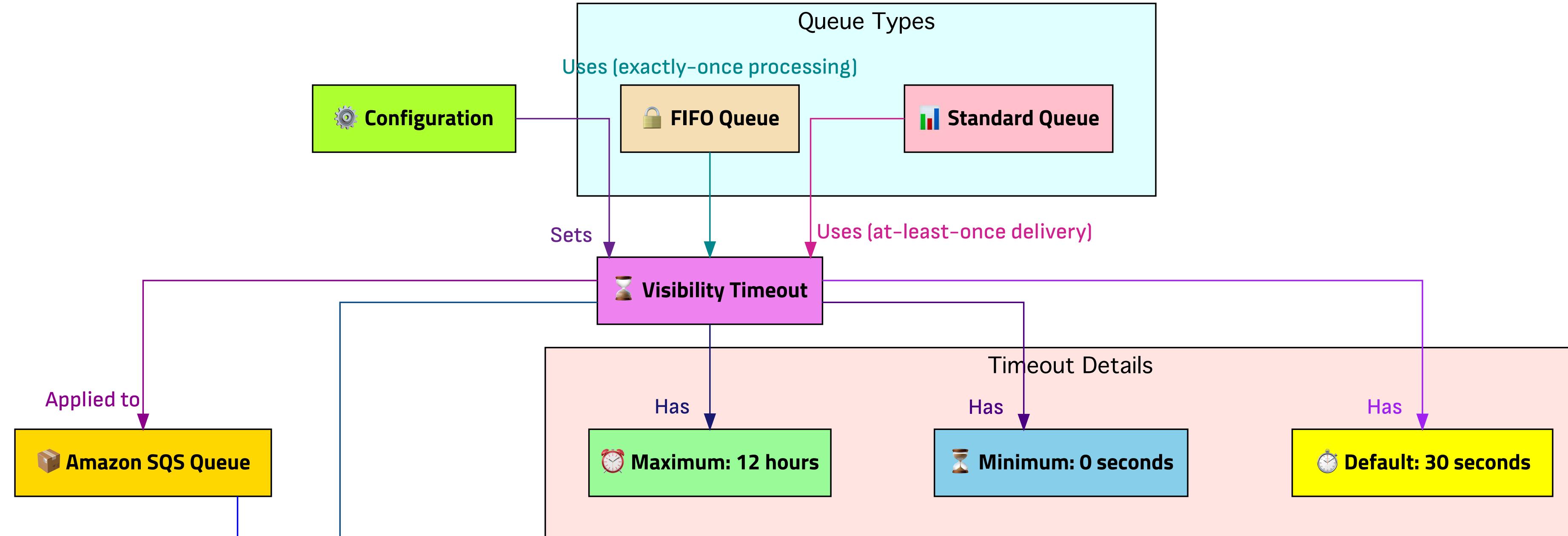


2. 🗑 Manual Deletion Requirement	
🌐 Distributed system nature	📡 Connectivity issues
⚠ No guarantee of message receipt	💻 Consumer application issues
👤 Consumer responsibility	✓ Receive and process message
	🗑 Delete message from queue

3. ⏳ Visibility Timeout Mechanism

- 🔒 Prevents other consumers from processing
- ⚡ Set immediately after message receipt

7. Amazon SQS Visibility Timeout



Sends message to Consumer

Prevents other consumers

Performs

Message Processing

Responsible for

Message Deletion

4. ⏳ Timeout Duration

Default: 30 seconds

Minimum: 0 seconds

Maximum: 12 hours

5. 🔒 Configuration

Using Amazon SQS console

6. 📈 Standard Queues

⚠ No guarantee against duplicate receipt

⟳ At-least-once delivery

7. 🔒 FIFO Queues

Producer: message deduplication ID

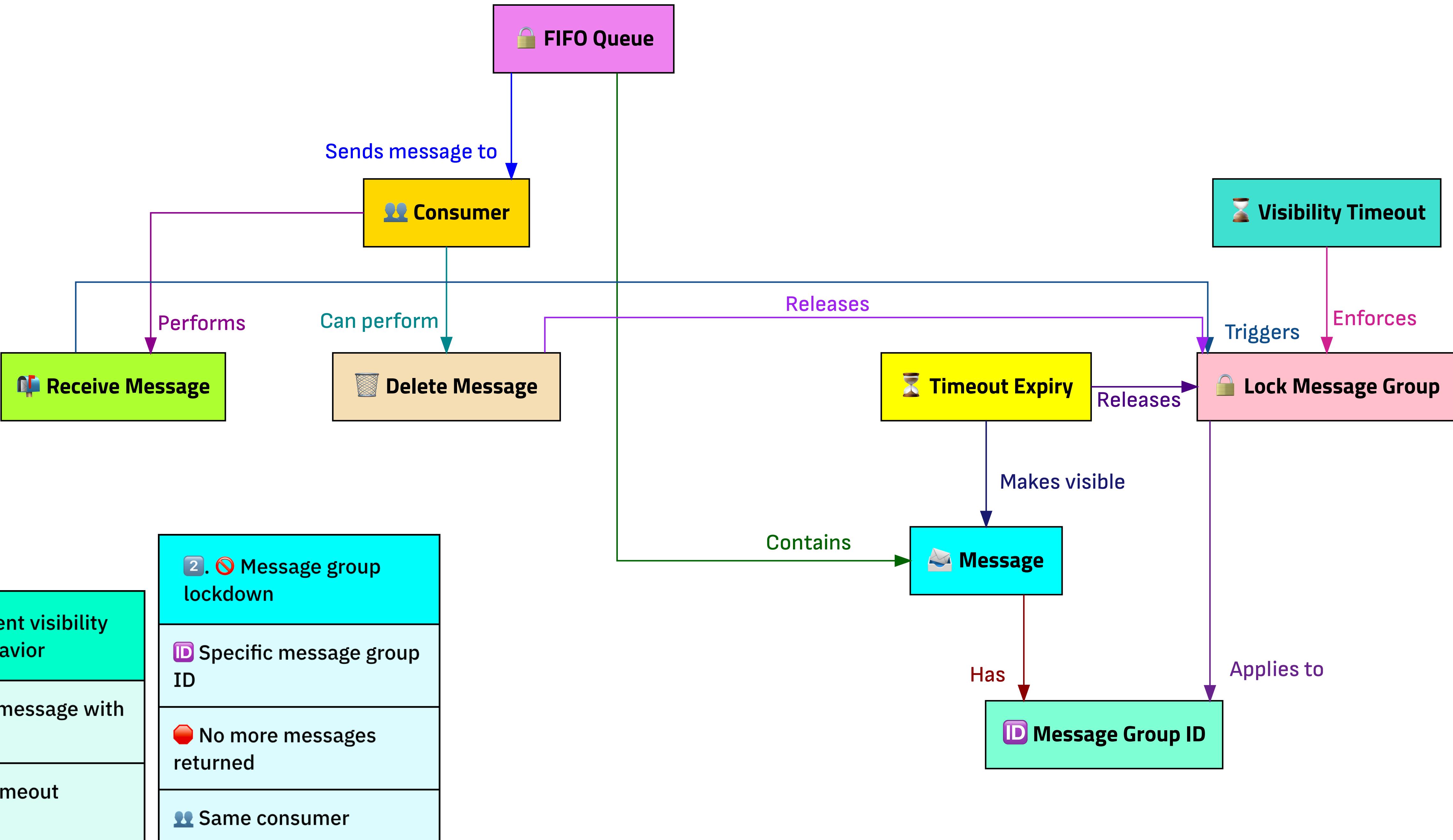
Consumer: receive request attempt ID

✓ Maintains message ordering

🚫 No duplicates introduced

⏳ Acknowledgement before timeout expiry

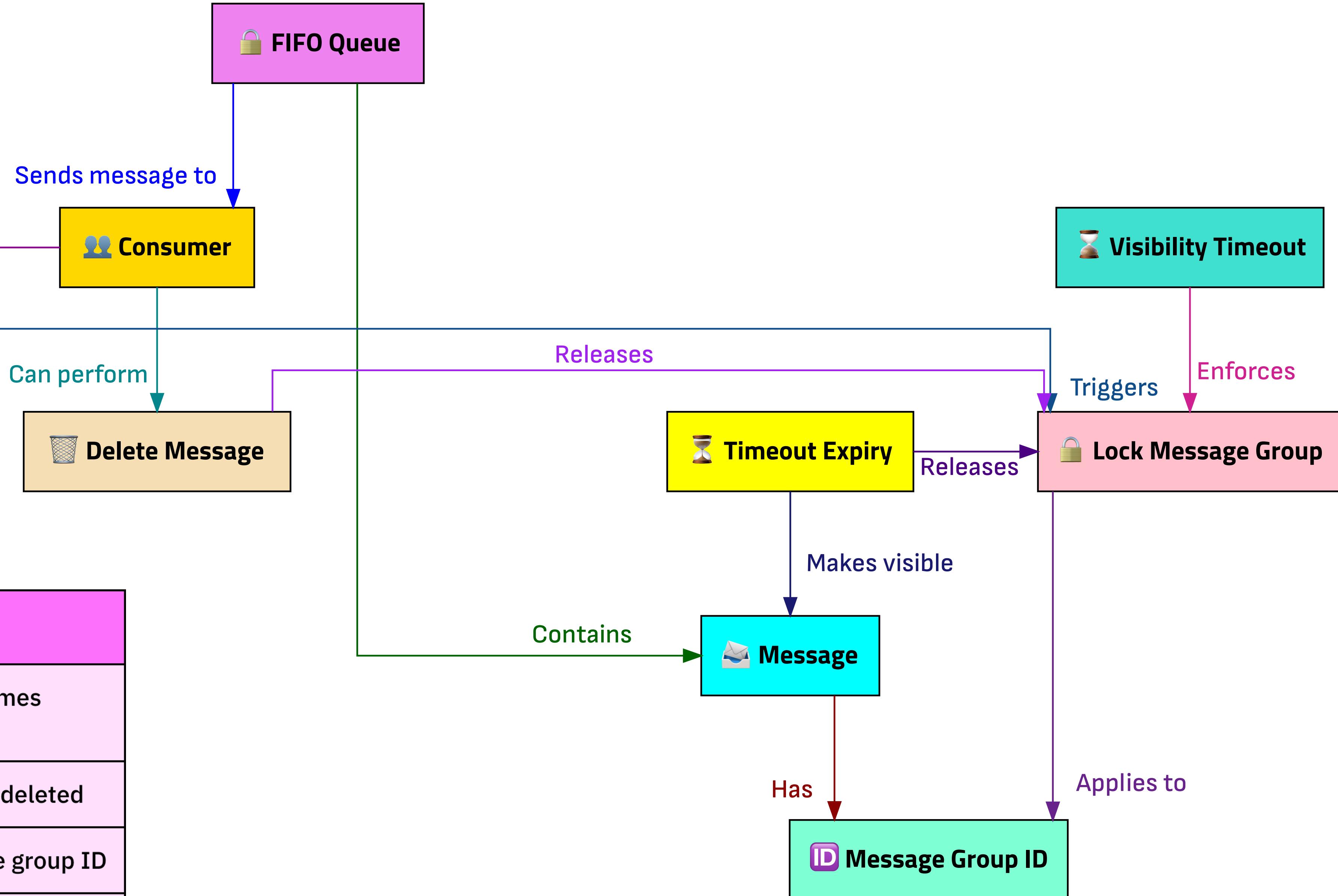
FIFO Queues: Message Group ID and Visibility



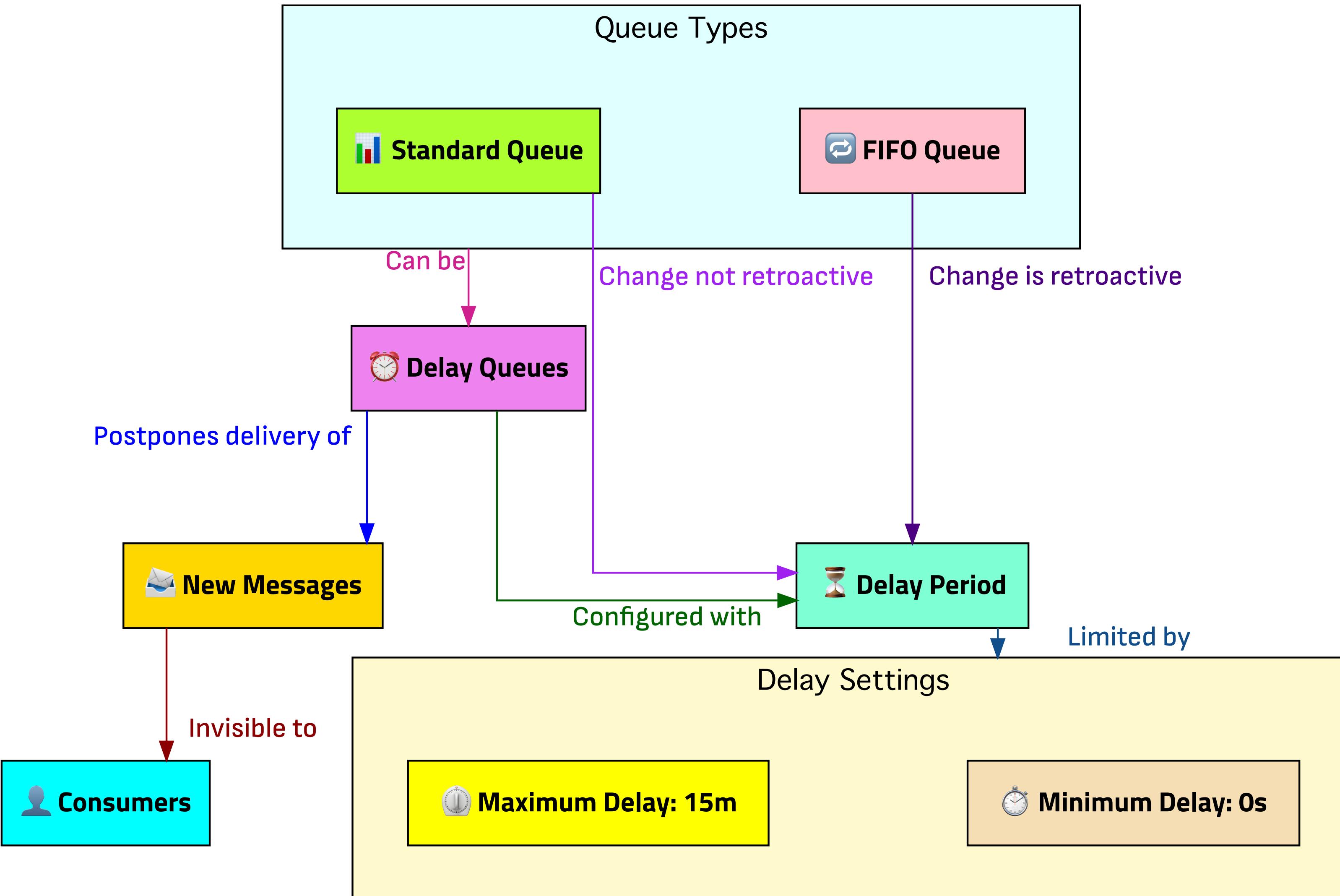
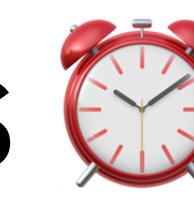
FIFO Queues: Message Group ID and Visibility

3.	Persistence of behavior
	Until message deletion
	By consumer
	From queue

4.	Alternative scenario
Visibility timeout expiry	Message becomes visible again Without being deleted
Subsequent messages	Same message group ID Can be returned to consumer



8. Amazon SQS Delay Queues



1. ⏳ Postpone message delivery
 - ⌚ Specified number of seconds
 - ⌚ Allows consumer processing time
 2. 🧑 Message invisibility
 - 🚫 Invisible to consumers
 - ⌚ During specified delay period
 3. ⏱ Default delay
 - ⌚ 0 seconds (minimum)
 - ⌚ Immediate availability
 4. ⏲ Maximum delay
 - ⌚ 15 minutes
 - ⌚ 900 seconds
5. 🔍 Configuration
 - 💻 Amazon SQS console
 - ⌚ Set delay period
6. 📈 Standard queues behavior
 - ⌚ Changing delay setting
 - 🚫 Not retroactive
 - ✉️ Doesn't affect existing messages
 7. 🔒 FIFO queues behavior
 - ⌚ Changing delay setting
 - ✓ Retroactive
 - ✉️ Affects existing messages



Delay Queues vs. Visibility Timeouts

1. ⏳⌚ Similarity in message unavailability
⏳ Specific time period
👀 Hidden from consumers

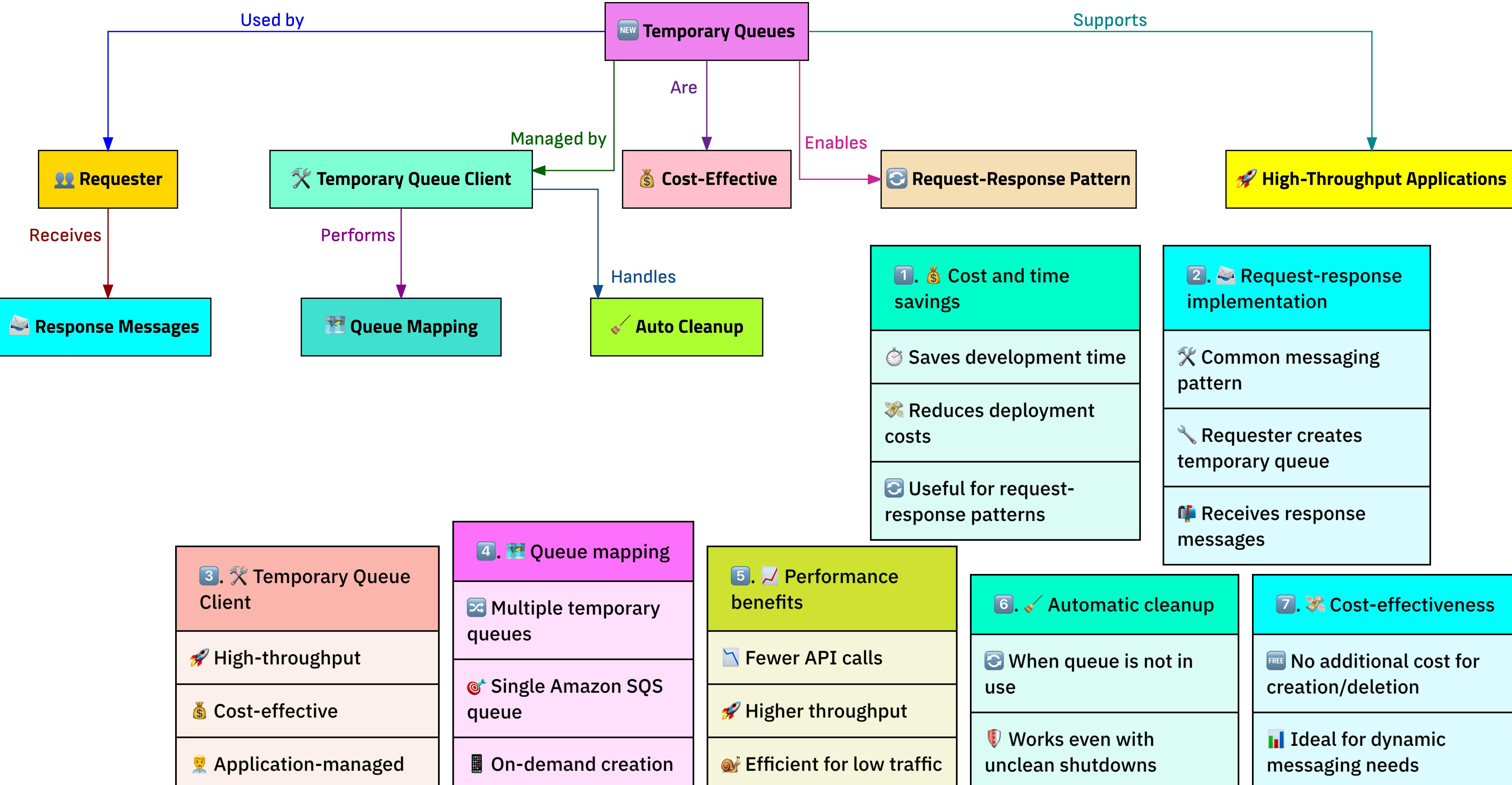
2. 🚶 Delay queues behavior
🆕 Message added to queue
🙈 Immediate invisibility
⌚ Duration of delay period

3. 👀 Visibility timeouts behavior
🔁 After message consumption
🙈 Hidden from other consumers
⌚ Duration of visibility timeout

4. ⏳⌚ Message timers
⌚ Individual message delays
🔀 Alternative to queue-wide delay

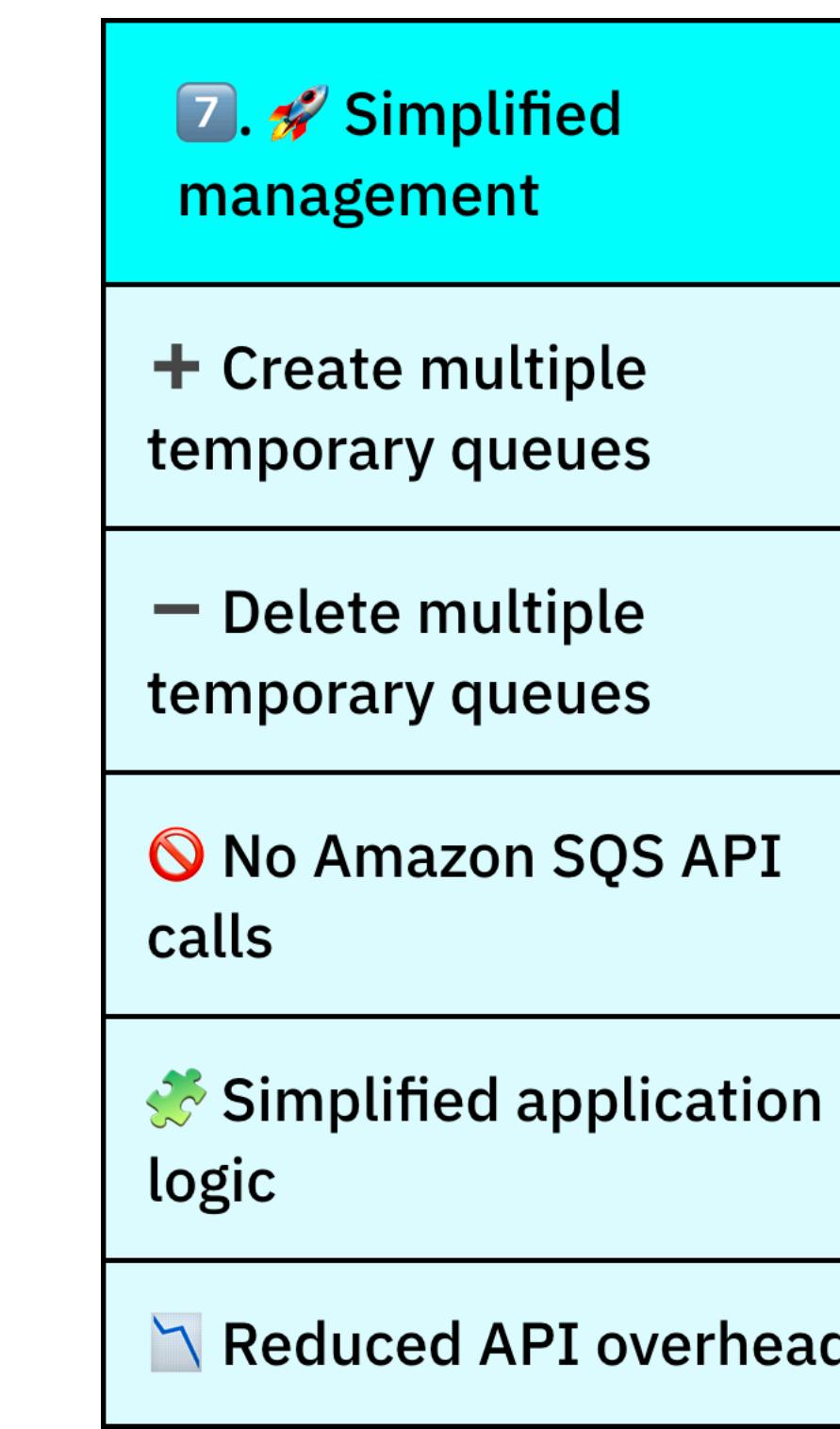
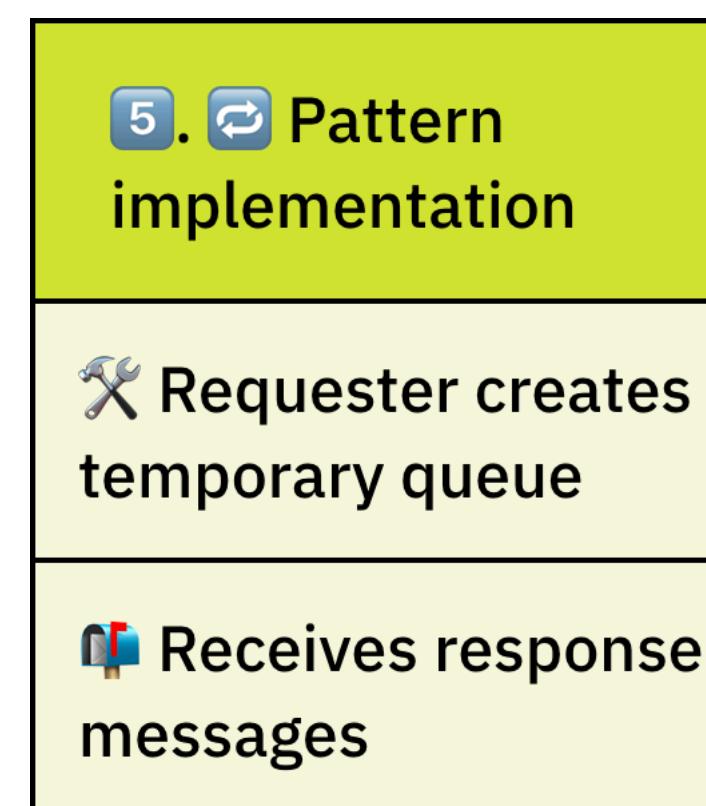
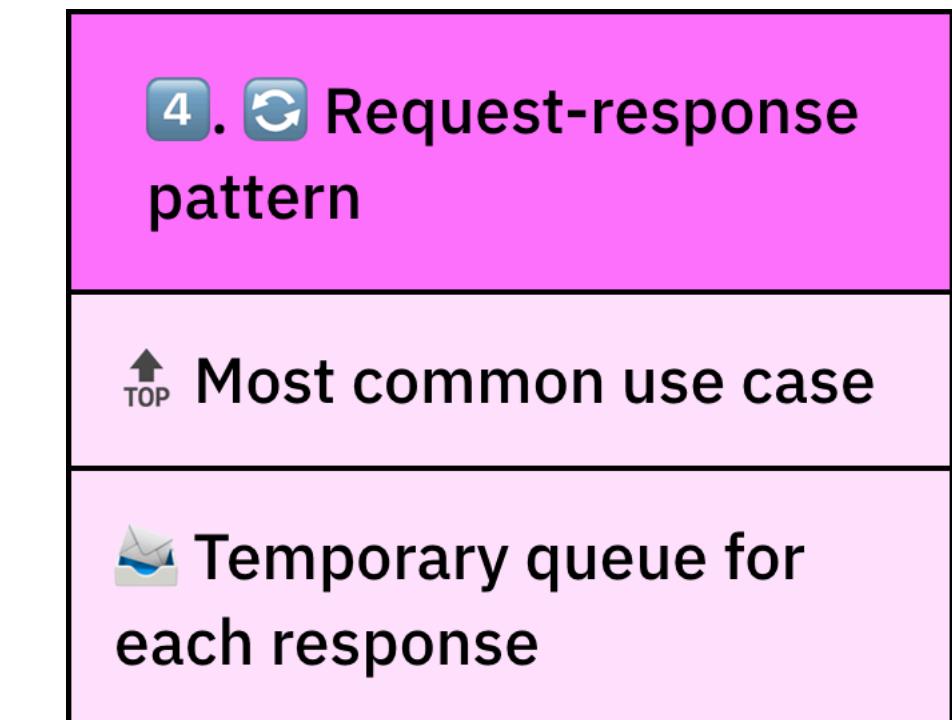
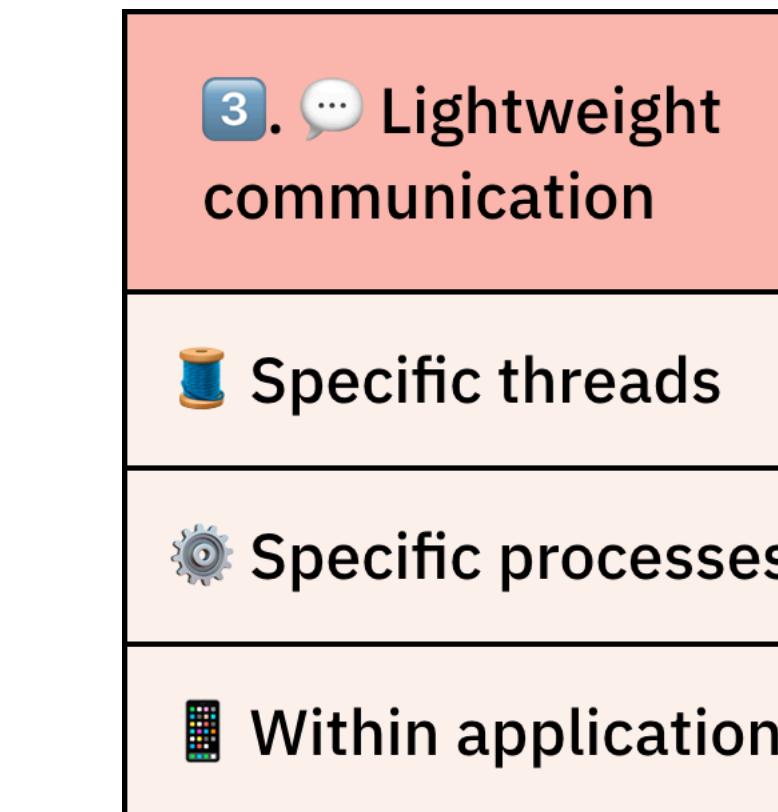
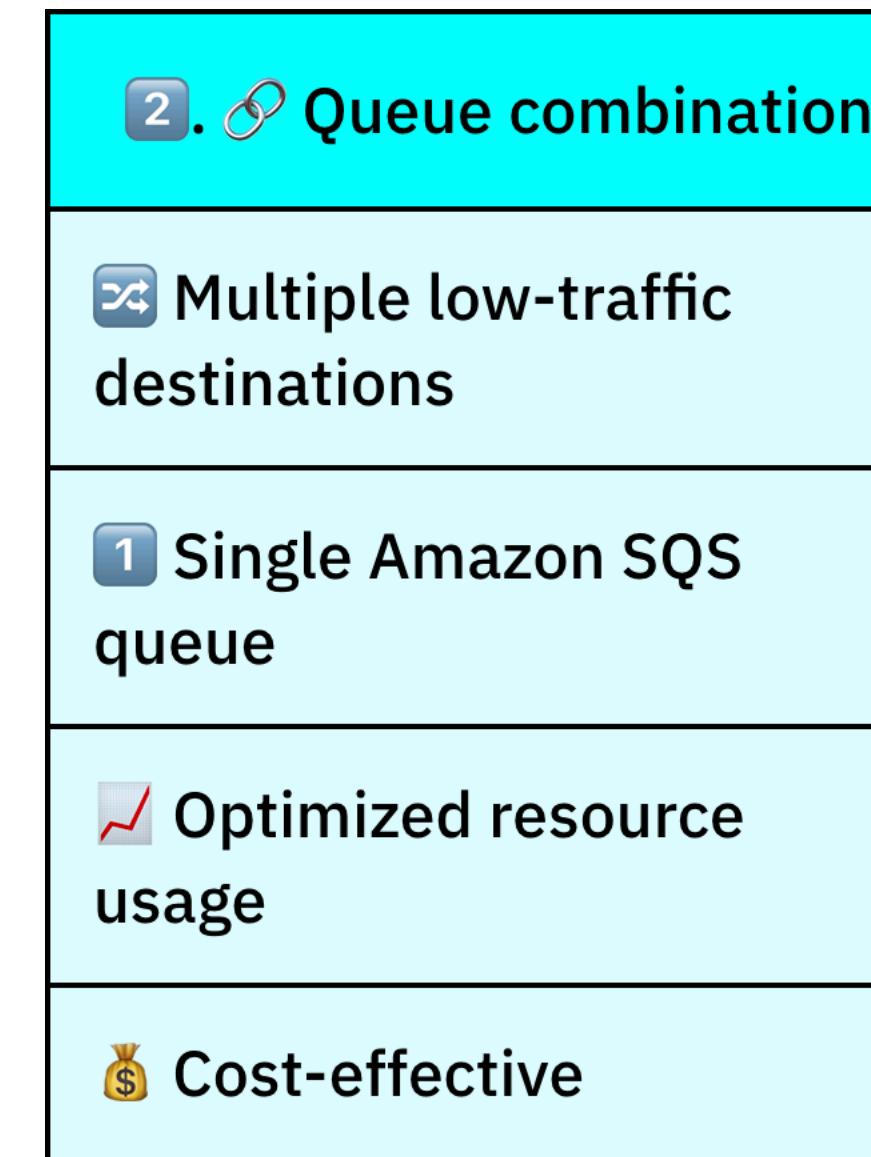
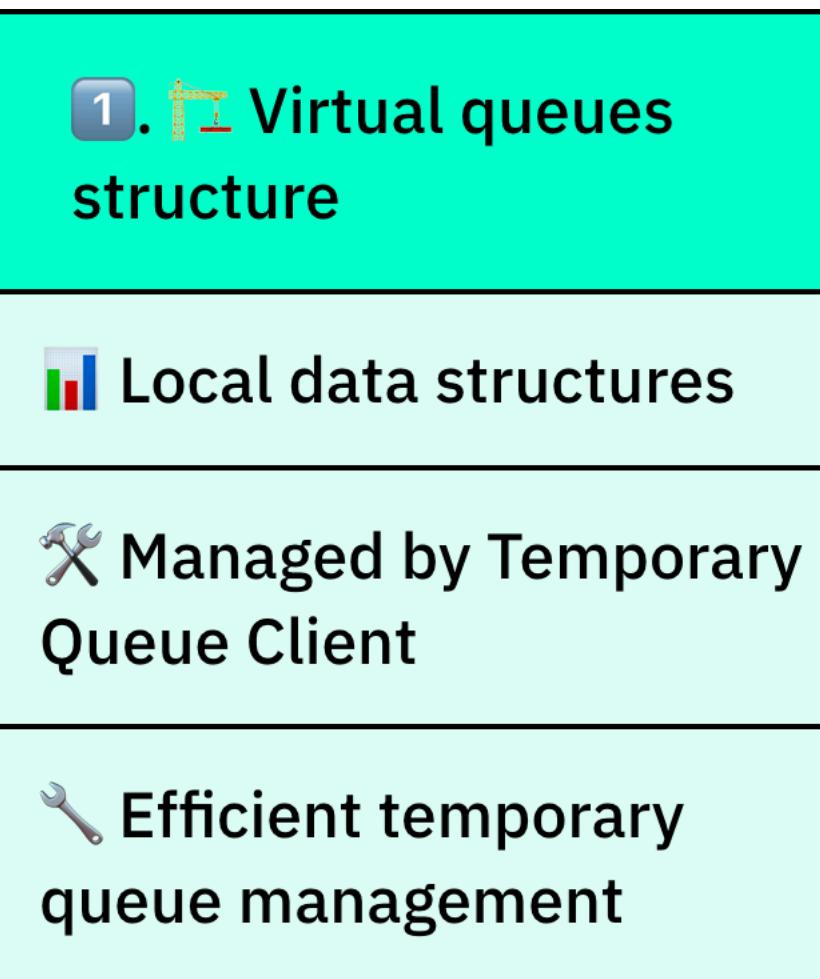
5. ⏲ DelaySeconds value
⌚ Message timer's value
🔁 Overrides queue's DelaySeconds
⌚ Applies to specific message

9. Amazon SQS Temporary Queues

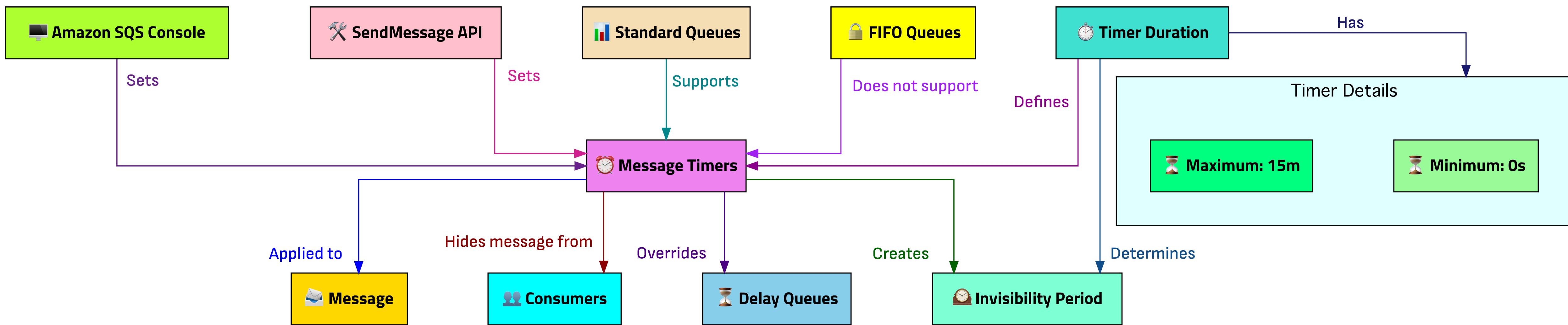




Virtual Queues and Request-Response Pattern

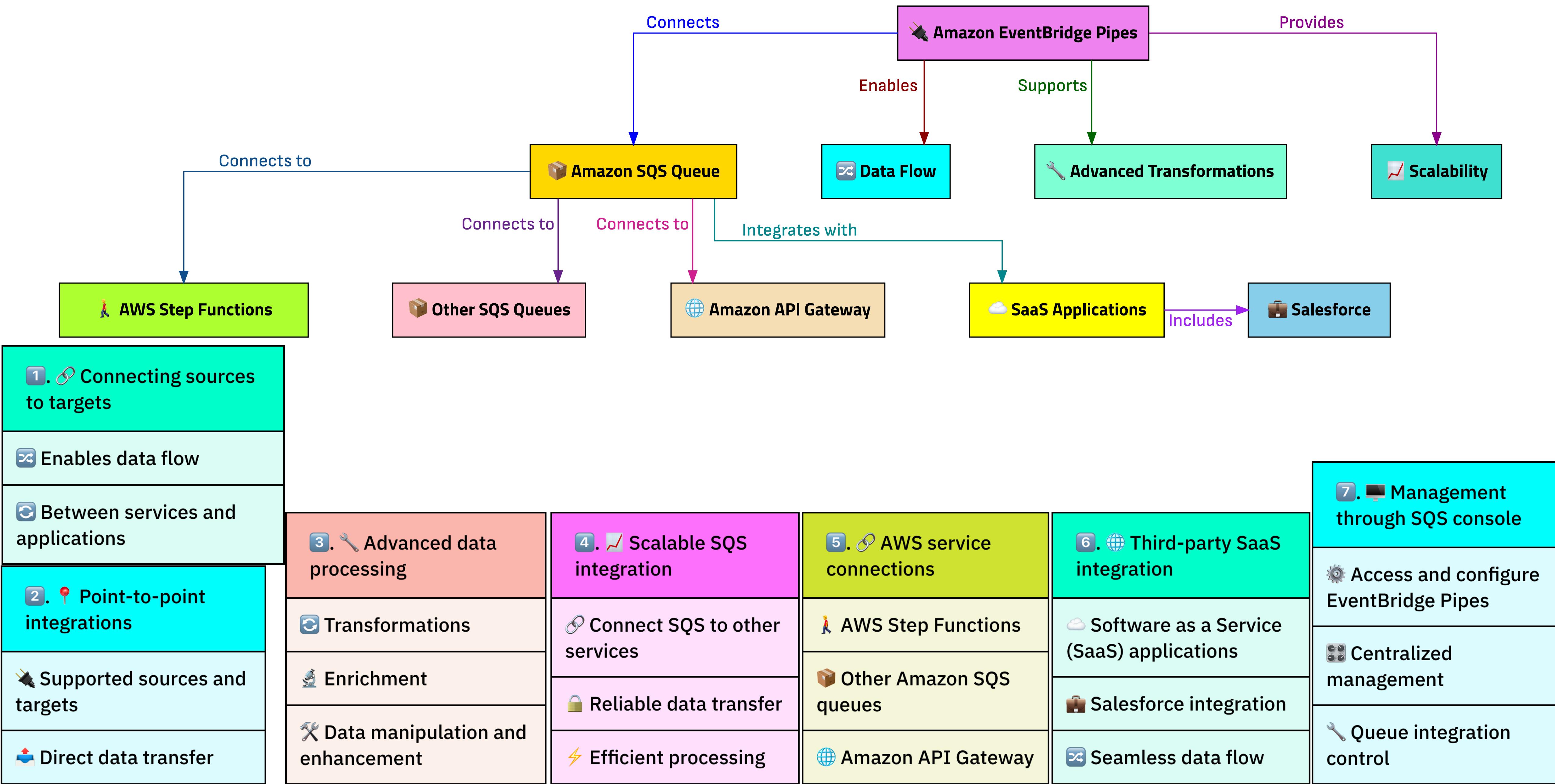


10. Amazon SQS Message Timers

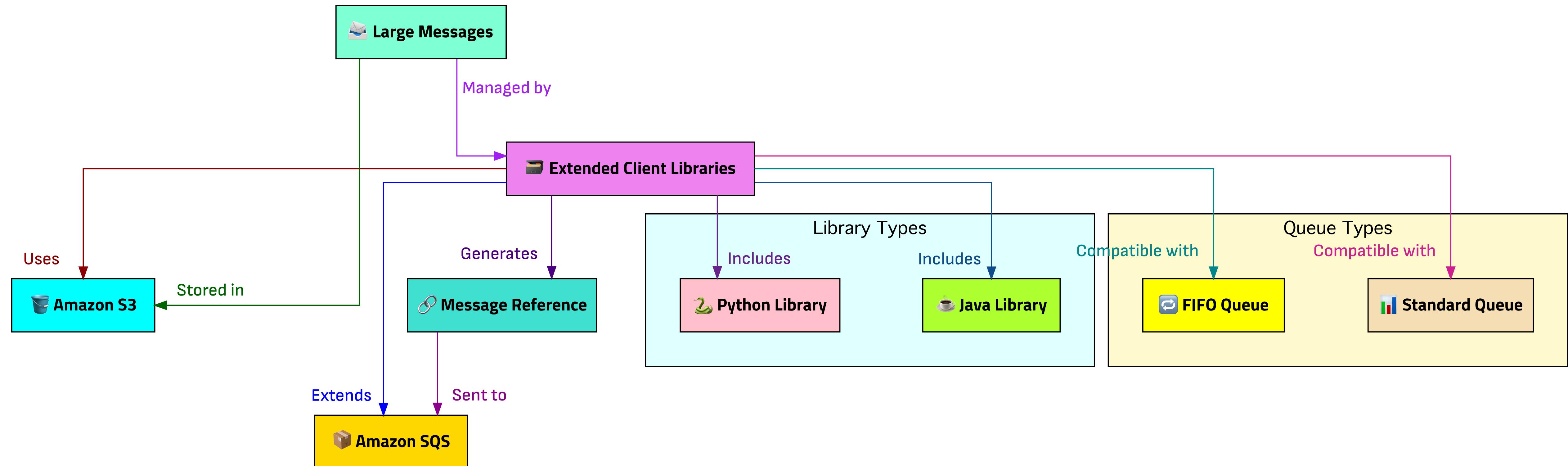


1. ⏰ Initial invisibility period	2. 📬 Message visibility	3. ⏱ Default delay			4. ⏲ Maximum delay	5. 🔧 Setting message timer	6. 💼 Queue compatibility	7. ⚙ Override behavior
⌚ Hidden from consumers	🚫 Not visible during timer period	⌚ 0 seconds (minimum)	⌚ 15 minutes	⌚ Immediately visible	⌚ 900 seconds	Amazon SQS console SendMessage API action	✓ Standard queues ✗ FIFO queues	⌚ Support individual message timers ✗ No support for individual message timers
⌚ Specified duration	⌚ After queue addition	⌚ 0 seconds (minimum)	⌚ 15 minutes	⌚ Immediately visible	⌚ 900 seconds			⌚ Overrides DelaySeconds value ⌚ Delay queue setting

11. Amazon EventBridge Pipes & Amazon SQS Integration



12. Managing Large Amazon SQS Messages



1. **Extended Client Libraries**

- Java library
- Python library
- Send and manage large messages

2. **Message Size Range**

- Minimum: 256 KB
- Maximum: 2 GB
- Larger than default SQS limit

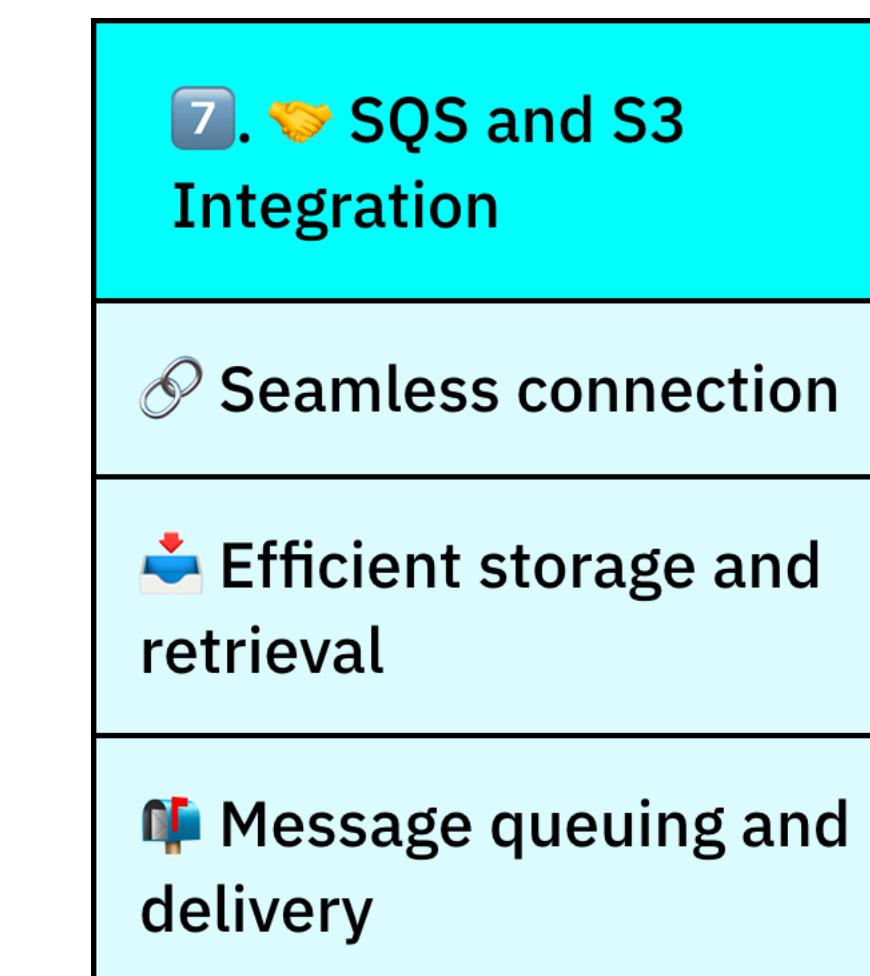
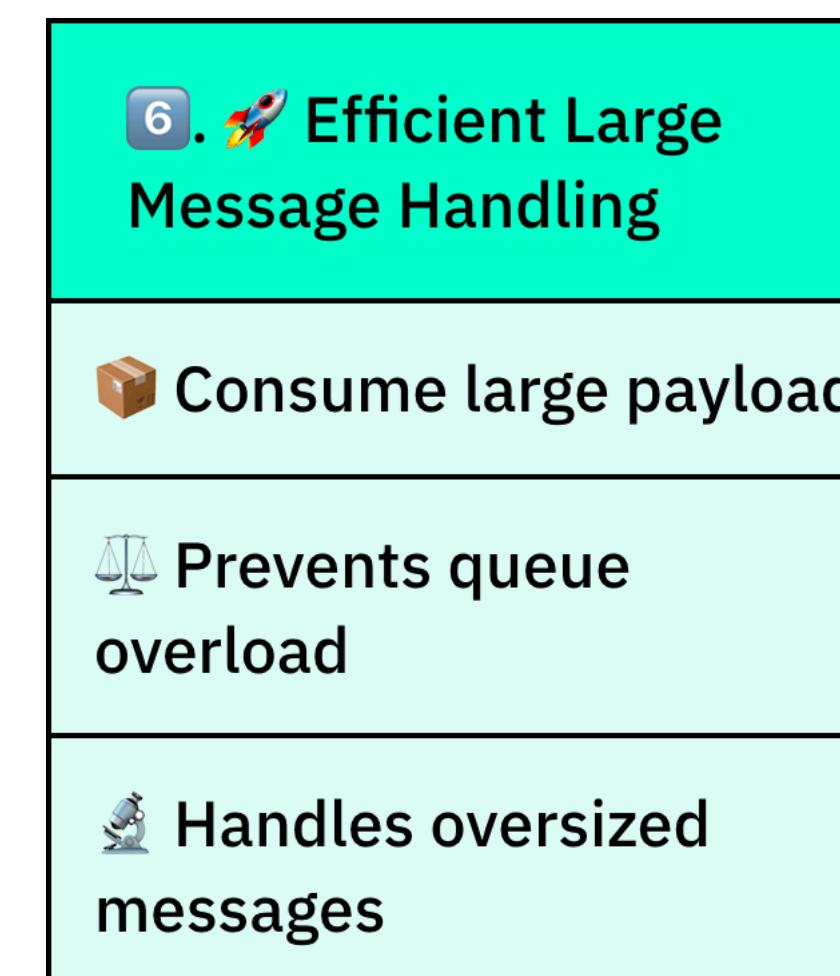
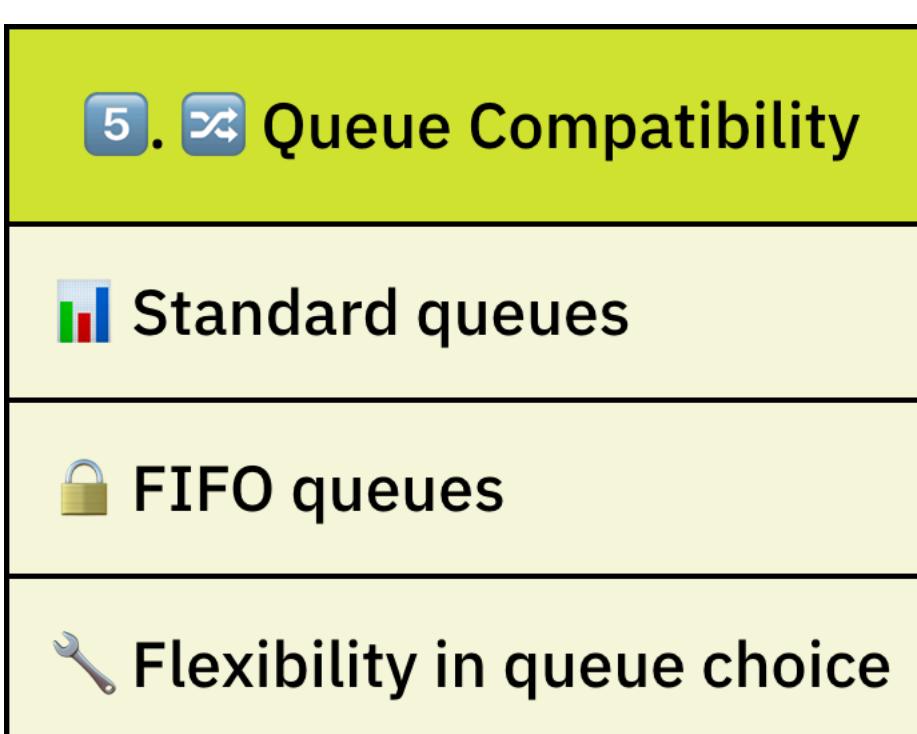
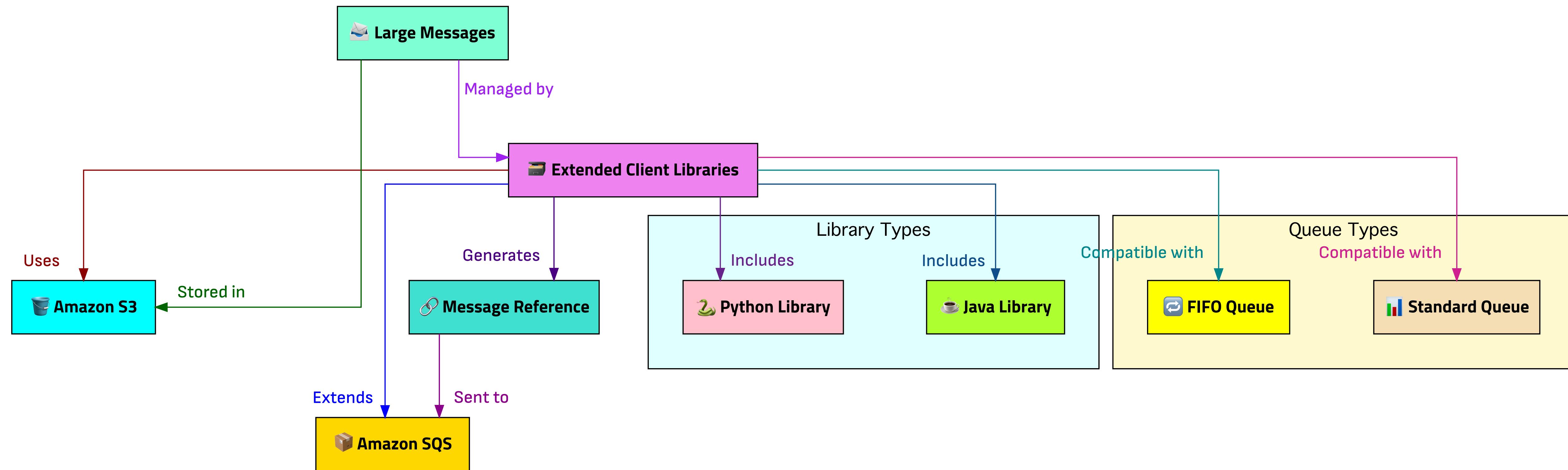
3. **S3 Storage Integration**

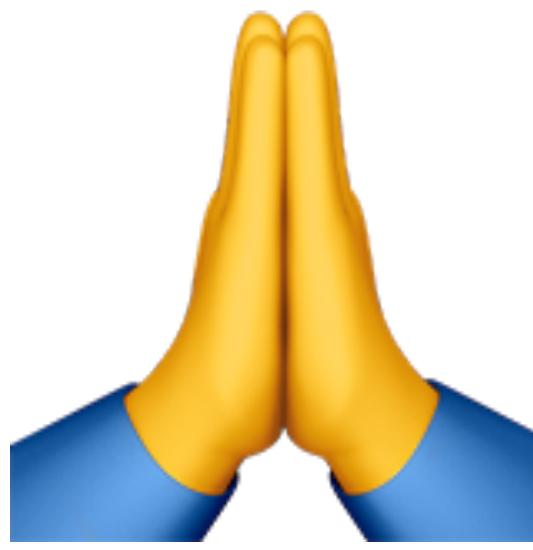
- Automatically saves large payloads
- Redirects from SQS to S3

4. **Message Reference System**

- Pointer to S3 object
- Sent in SQS queue
- Allows payload retrieval

12. Managing Large Amazon SQS Messages





**Thanks
for
Watching**