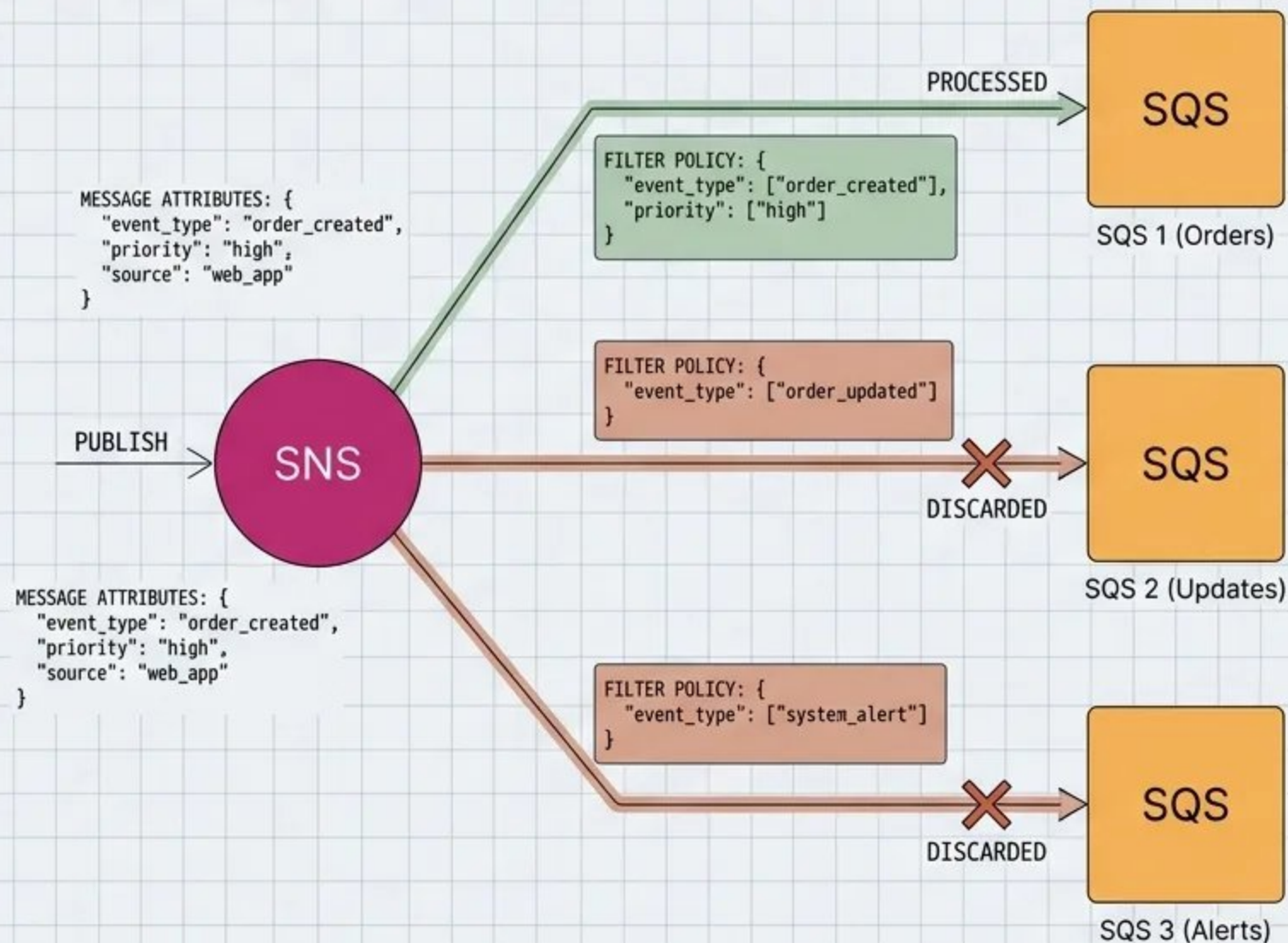


Roteamento Inteligente na AWS

Dominando SNS MessageAttributes e Filter Policies

Como filtrar mensagens antes do SQS, reduzir custos em escala e desacoplar sua arquitetura de verdade.

ARCHITECTURE DIAGRAM: SNS TO SQS FILTERING



O Problema do Fan-Out Padrão

Sem filtros, o modelo Pub/Sub cobra um preço alto pelo desperdício.

Por padrão, quando um tópico SNS tem várias filas SQS assinando, cada fila recebe todas as mensagens publicadas. O consumidor precisa receber, ler e descartar o que não é dele.

Você paga três vezes pela ineficiência:

1. Transferência de dados do SNS para o SQS.
2. Requisições no SQS (Envio, Recebimento, Exclusão).
3. Ciclos de CPU do seu consumidor processando lixo.

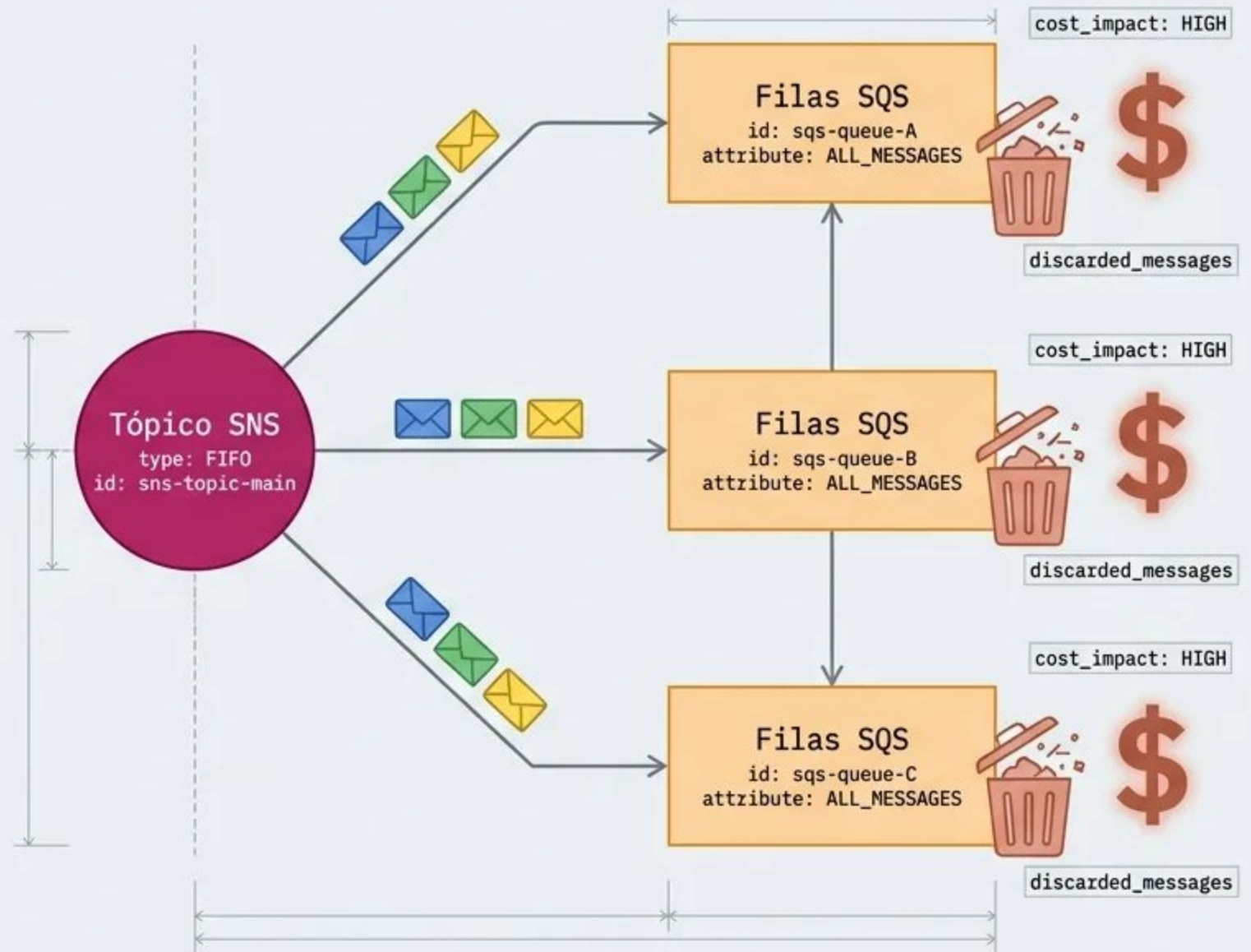
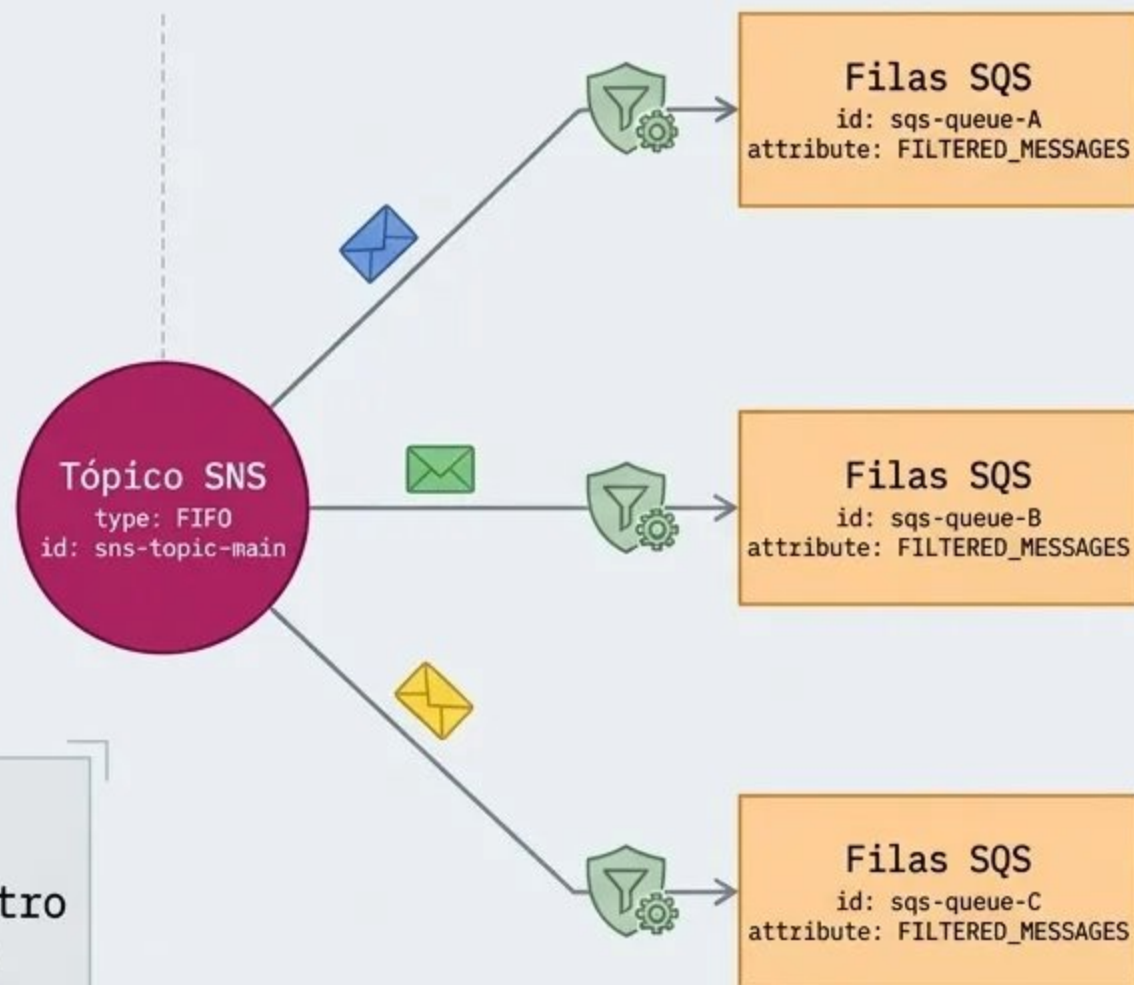


Diagram 1.1: Unfiltered Fan-Out Data Flow & Cost Impact

A Solução: A Válvula Inteligente (Filter Policy)

A Filter Policy delega o roteamento para a infraestrutura, entregando apenas o que interessa.



Fica na Assinatura

O SNS toma a decisão de entrega antes da mensagem tocar na fila.

Entrega Seletiva

O que não passa no filtro JSON é silenciosamente descartado pelo SNS. Você não é tarifado no SQS.

Anatomia do Ecossistema

O filtro pertence à Assinatura (Subscription), não ao Tópico nem à Fila.



Tópico

O canal de broadcast onde o produtor publica os eventos.

Assinatura

O vínculo entre o Tópico e o destino (Fila). É o único lugar onde a regra de roteamento é armazenada. Sem política, a assinatura recebe tudo.

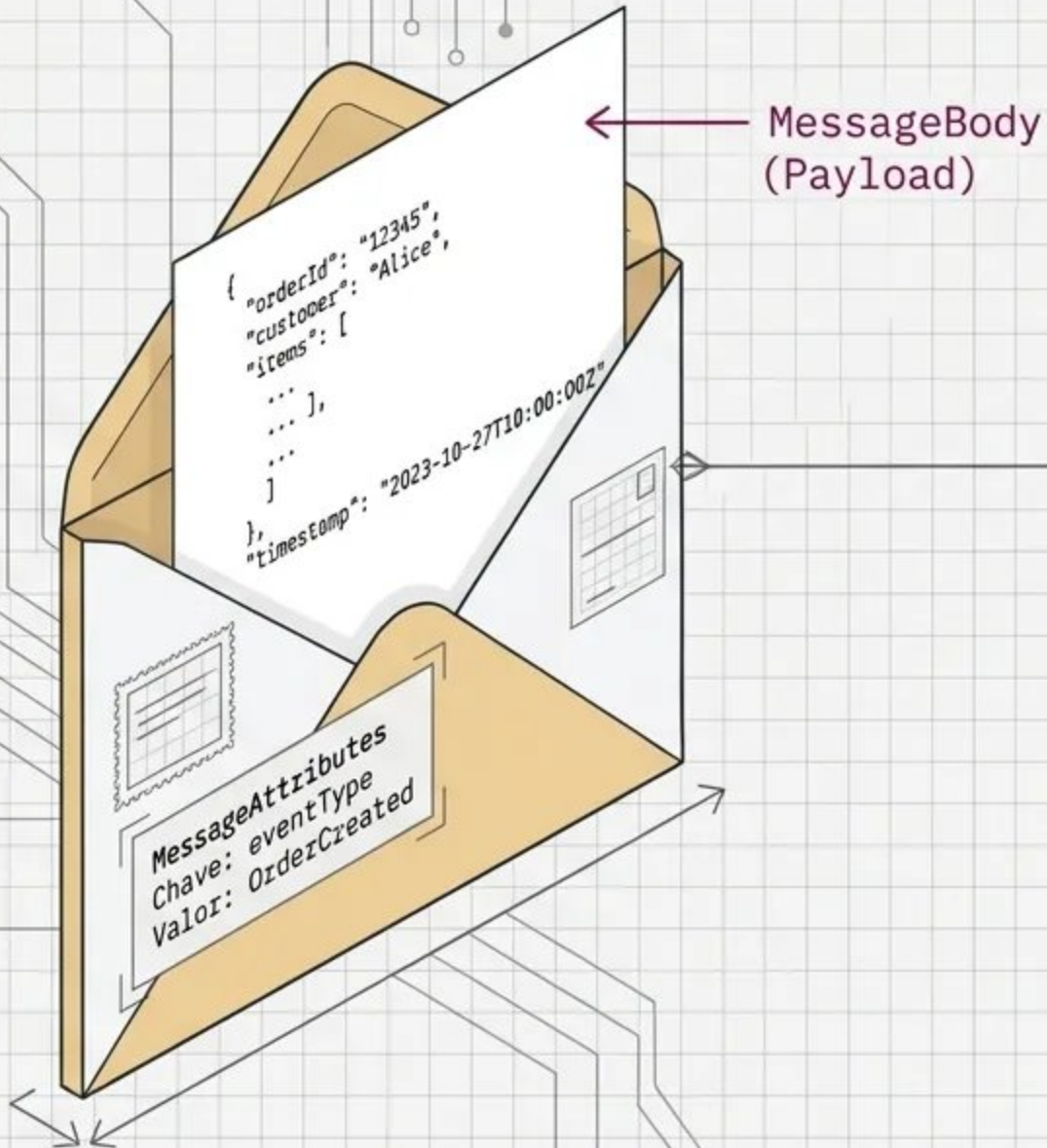
Fila

O buffer final. Quando o evento chega aqui, ele já foi pré-aprovado pela regra.

O Envelope da Mensagem

Atributos vs. Payload

Para rotear com eficiência, extraia os metadados do corpo da mensagem.



MessageAttributes

São metadados opcionais que viajam junto com a mensagem, mas do lado de fora do corpo.

Eles são lidos instantaneamente pelo motor do **SNS** sem precisar fazer o parsing (deserialização) de todo o seu evento de negócio.

Tipos Suportados: String, String.Array, Number e Binary. (Nota: O filtro SNS ignora atributos binários).

O Lado do Produtor: Adicionando Atributos

Injetando as etiquetas de roteamento na origem.

Java / Spring Cloud AWS

```
snsTemplate.convertAndSend(  
    topicArn,  
    event,  
    // Corpo: serializado em JSON  
    Map.of("eventType", "OrderCreated")  
    // Header vira MessageAttribute  
);
```

AWS CLI

```
aws sns publish \  
    --topic-arn arn:aws:sns:... \  
    --message '{"id": "123"}' \  
    --message-attributes  
        '{"eventType": {"DataType": "String",  
            "StringValue": "OrderCreated"}}'
```



Alerta para Spring Cloud AWS:

A autoconfiguração do SnsTemplate converte headers automaticamente em MessageAttributes do SNS.

Cuidado: Evite usar `send(topicArn, message)` passando objetos complexos, pois ele publicará o `toString()` do objeto em vez de um JSON estruturado. Sempre serialize o payload.

O Lado do Consumidor: A Catraca Lógica

A Filter Policy é um JSON simples anexado à Assinatura.

```
{  
  "eventType": ["OrderCreatedEvent"]  
}
```



Seja via Console, AWS CLI ou Infraestrutura como Código (IaC), o atributo `FilterPolicy` dita a regra do jogo.

No exemplo acima, a fila correspondente a essa assinatura receberá exclusivamente mensagens cujo atributo `eventType` seja exatamente `OrderCreatedEvent`. Tudo que for diferente é barrado na porta.

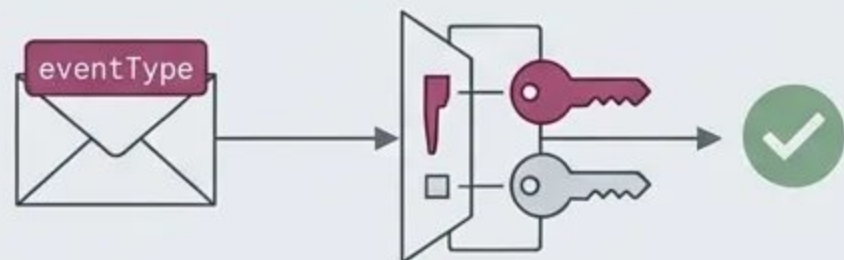
O Motor de Avaliação do SNS

Como o SNS interpreta o seu JSON em microssegundos:



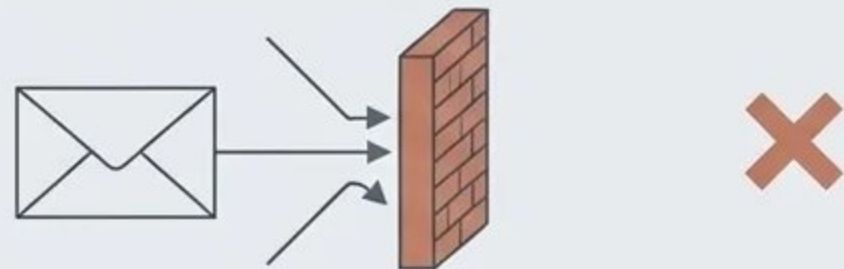
Chaves Diferentes = AND Lógico

Se a política tem chaves diferentes (ex: eventType E region), a mensagem precisa satisfazer todas elas.



Mesmo Array = OR Lógico

Se os valores estão no mesmo array (ex: ["OrderCreated", "OrderUpdated"]), basta casar com um deles.

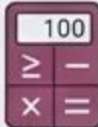


Atributo Ausente = DROP

Se a regra exige um atributo e a mensagem chega sem ele, ela é sumariamente descartada (exceto se a regra for "exists": false). Atributos extras na mensagem são ignorados.

“Cheat Sheet” de Operadores (Matchings)

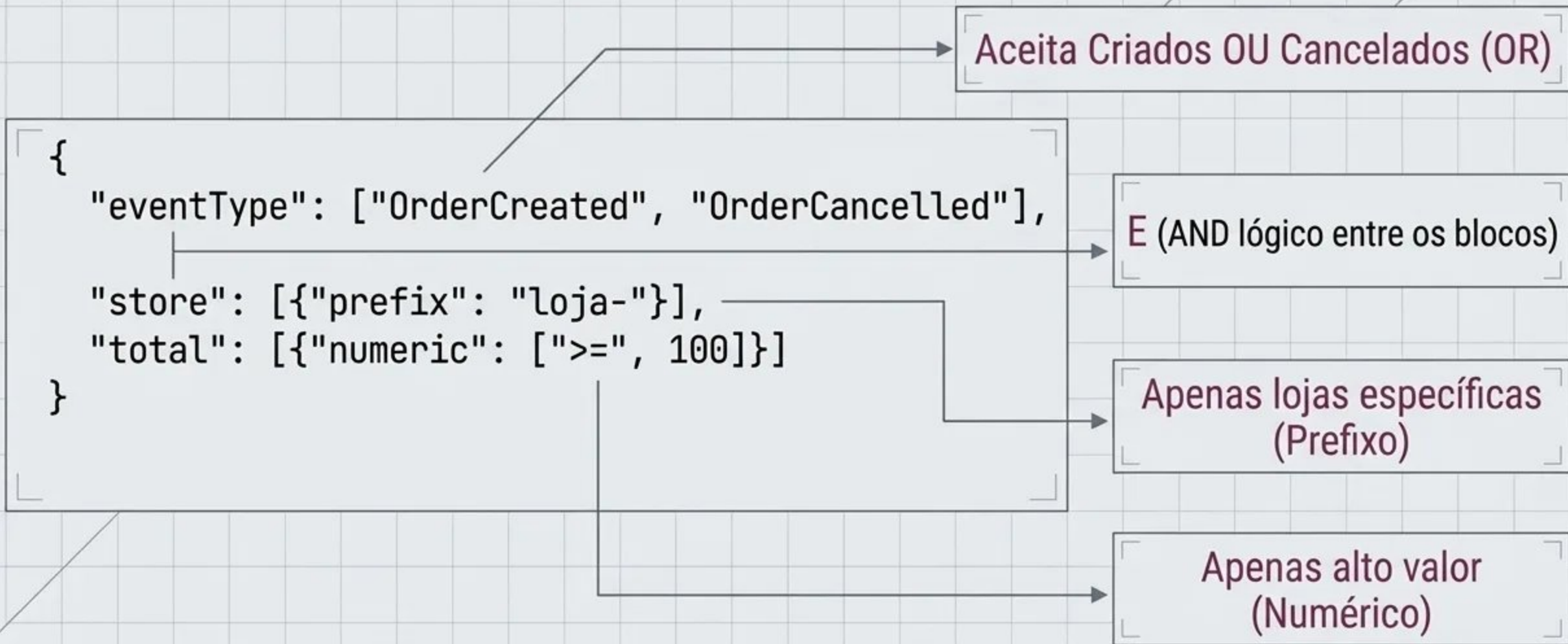
Além da igualdade exata de strings, o SNS oferece operadores nativos poderosos.

Sintaxe JSON (Operador)	Critério de Sucesso
<code>[{"anything-but": ["OrderShipped"]}]</code>	 Valor não é o listado.
<code>[{"prefix": "loja-"}]</code>	 Começa com o texto.
<code>[{"equals-ignore-case": "order"}]</code>	<code>a=A</code> Ignora maiúsculas.
<code>[{"numeric": [">=", 100]}]</code>	 Maior ou igual a 100.
<code>[{"exists": true}]</code>	 Atributo está presente.

Dica: A comparação padrão de strings diferencia maiúsculas de minúsculas.
Use `equals-ignore-case` se não tiver controle estrito sobre o formato do produtor.

Roteamento Complexo na Prática

Combinando operadores, você tira regras de negócio complexas do código da sua aplicação e as coloca direto na infraestrutura.



Matriz de Decisão: Onde o SNS deve procurar?

O atributo FilterPolicyScope define se a regra avalia o envelope ou o conteúdo.

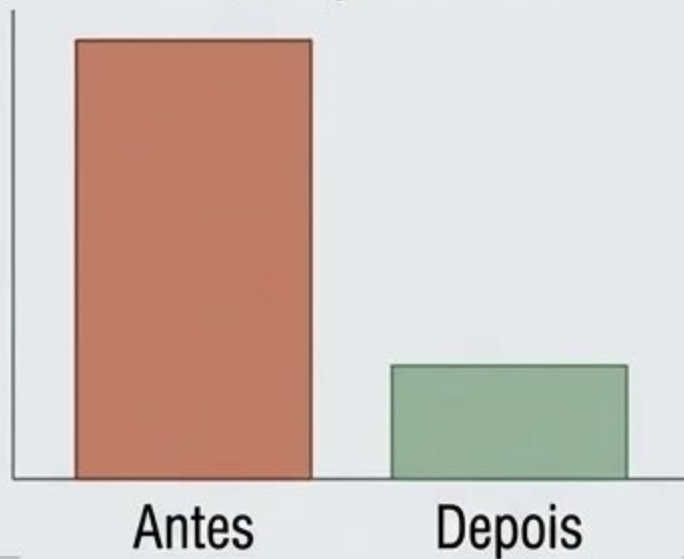
Dimensão	MessageAttributes (Padrão Vencedor)	MessageBody
Custo	Gratuito	Cobrado por GB analisado
Estrutura	Linear (apenas atributos raiz)	Suporta JSON aninhado
Acoplamento	Desacoplado do payload	Fortemente acoplado ao formato
Atributos Binários	Ignorados pelo motor	N/A
Quando usar?	Na imensa maioria dos casos.	Quando você não controla o produtor e ele não envia atributos.

Filtrar pelos Atributos é quase sempre a melhor escolha arquitetural: é gratuito e mantém a regra de roteamento agnóstica ao payload do negócio.

O Impacto Arquitetural (Business Case)

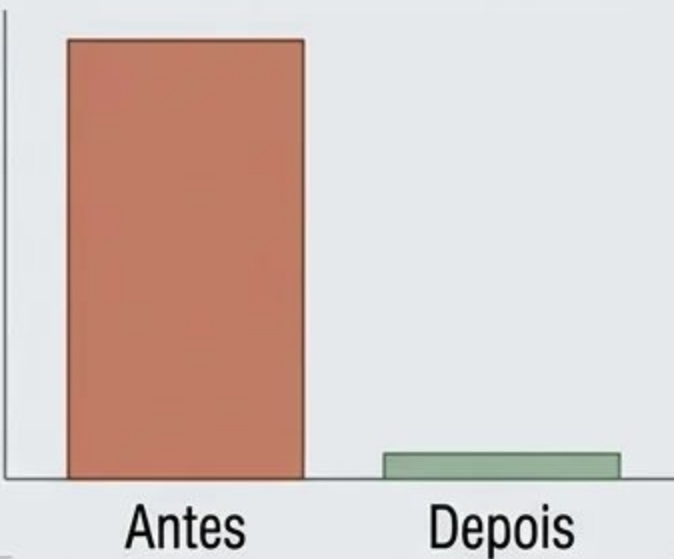
Adicionar Filter Policies não é apenas uma otimização técnica; é uma alavanca direta de FinOps.

Requisições SQS



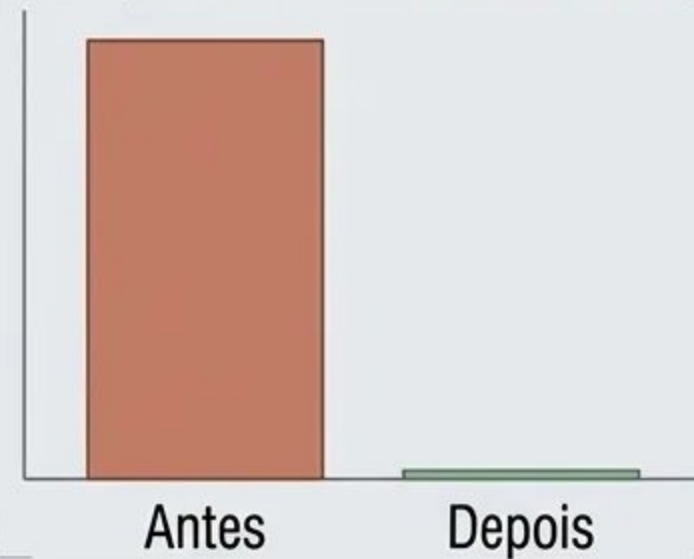
Queda massiva. (Cobrado por envio, recebimento e exclusão).

Transferência de Dados



Redução radical no volume de GBs de dados do SNS para o SQS.

Tempo de CPU (Consumidor)

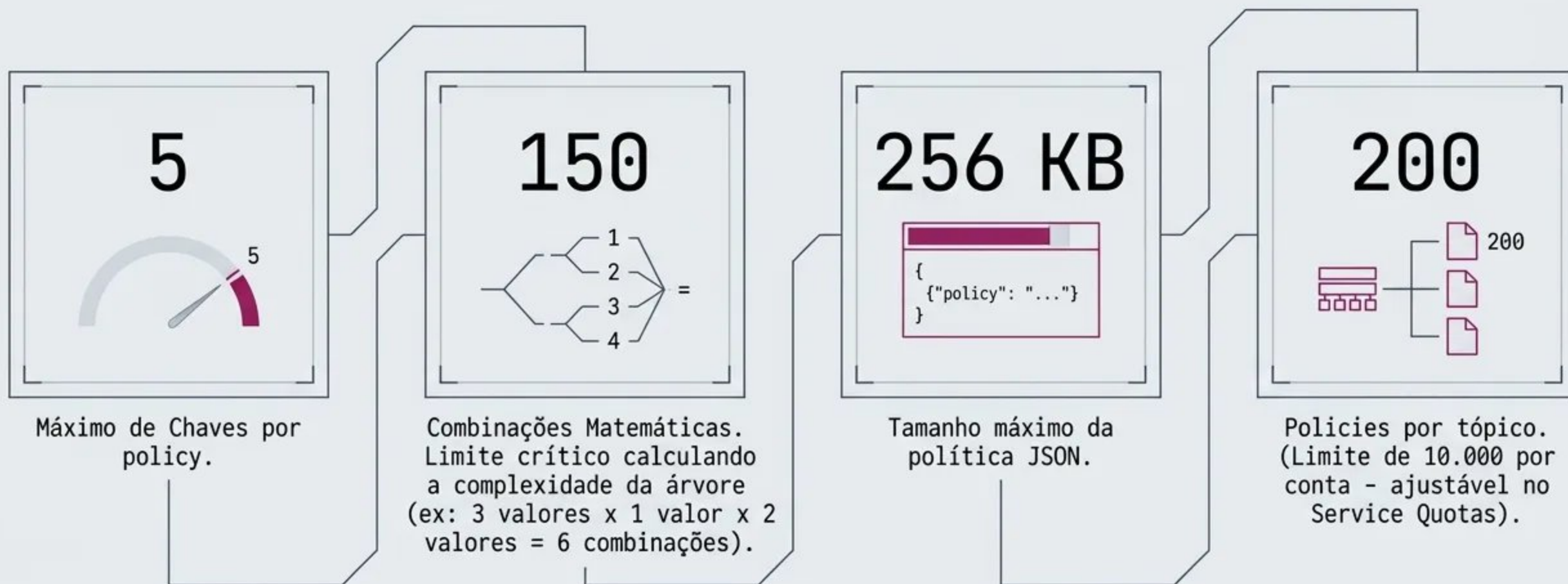


Zero processamento inútil no downstream.

Ao impedir que mensagens irrelevantes alcancem o SQS, você corta a fatura da AWS em múltiplas frentes simultaneamente. E o mais importante: preserva o compute dos seus microsserviços exclusivamente para eventos que geram valor para o negócio. É o desacoplamento clássico do pub/sub, mas com eficiência máxima.

Painel de Limites (Quotas de Serviço)

Planeje sua topologia respeitando os limites da AWS.



A restrição mais comum é o limite de combinações matemáticas (150). O motor do SNS multiplica a quantidade de valores de cada chave para calcular a complexidade da árvore de decisão. Mantenha suas políticas granulares e focadas.

A Zona de Perigo: Armadilhas Operacionais

Cenários críticos de falha no roteamento de mensagens.



Consistência Eventual

Criar ou alterar uma política leva até 15 minutos para entrar em vigor. Não entre em pânico (nem faça rollback) logo após o deploy se algumas mensagens ainda passarem.



Limites do Raw Message Delivery

Com esse modo ligado, a fila aceita no máximo 10 atributos. Mensagens excedentes são descartadas.

Cuidado: O Spring Cloud AWS injeta os headers "id" e "timestamp" automaticamente, consumindo 2 da sua cota.

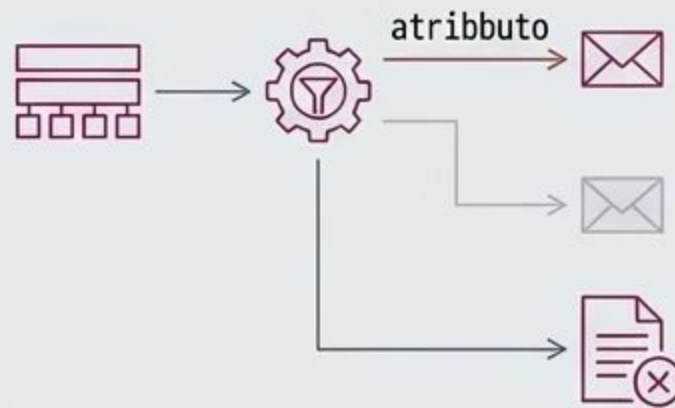


10
Max



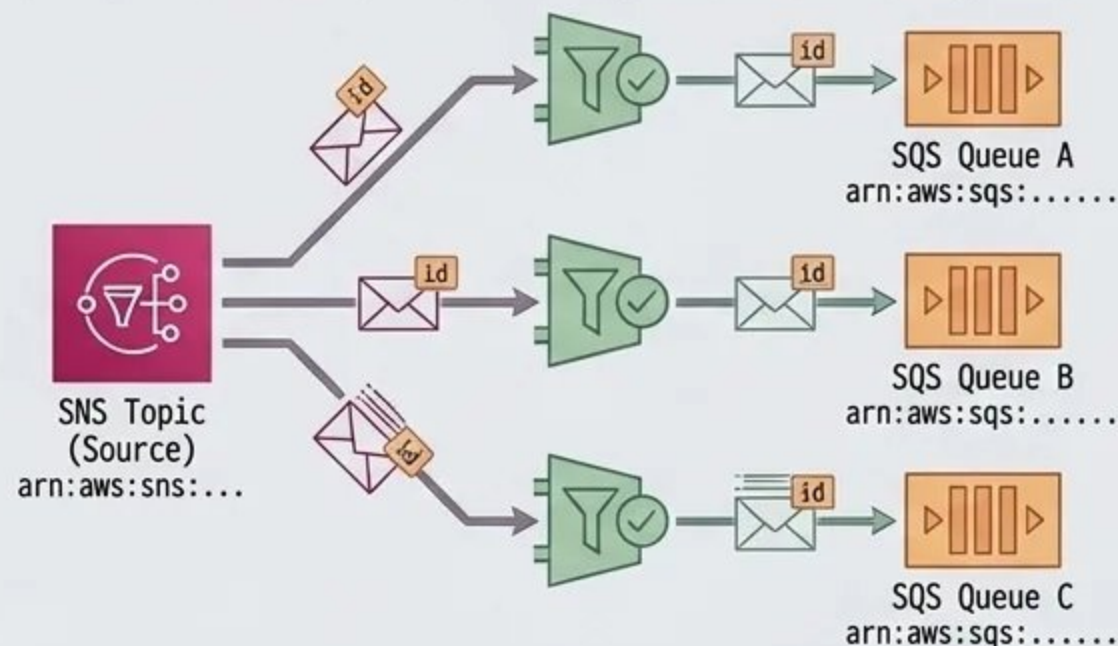
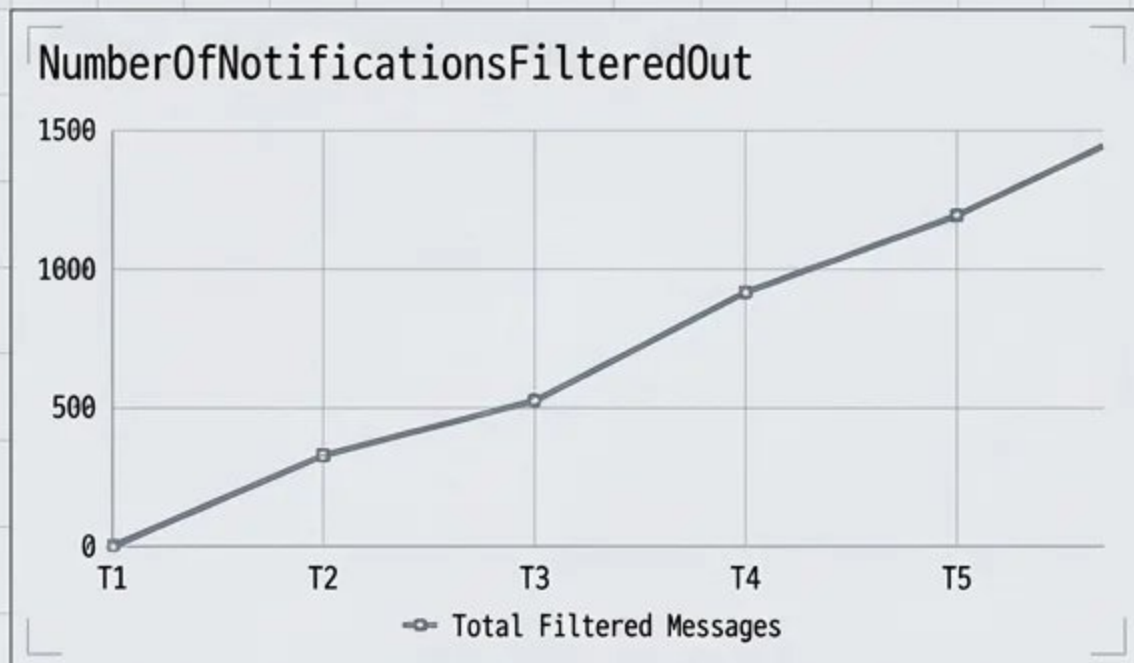
Erros Silenciosos

Um erro de digitação no nome do atributo na policy faz a mensagem ser descartada em silêncio absoluto. A aplicação não gera logs de erro, pois o filtro ocorre na infraestrutura.



Observabilidade e o Funil Inteligente

Filtros barrando mensagens não são erros; são o sistema funcionando perfeitamente.



Para garantir que suas políticas estão operando corretamente e não barrando eventos legítimos devido a typos, monitore implacavelmente a métrica **NumberOfNotificationsFilteredOut** (e suas variantes **-NoMessageAttributes** e **-InvalidAttributes**) no CloudWatch.



Takeaway Final: Filtros SNS protegem seus consumidores de sobrecarga e sua fatura AWS de desperdício. Marque com atributos, monitore as métricas e filtre na origem.