
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

NFX302

Accelerate insights using AWS SDK instrumentation

Nick Siow

Sr. Infrastructure Security Engineer
Netflix

Scott Pack

Sr. Cloud Security Engineer
Netflix

Prateek Sharma

Senior Solutions Architect
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Speaker and team intros

Nick Siow and Scott Pack

Cloud infrastructure security at Netflix

- Account architecture & provisioning
- Guidance & consulting
- Secure resource provisioning
- Vulnerability & misconfiguration



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The question

Which AWS resources does a given application interact with?

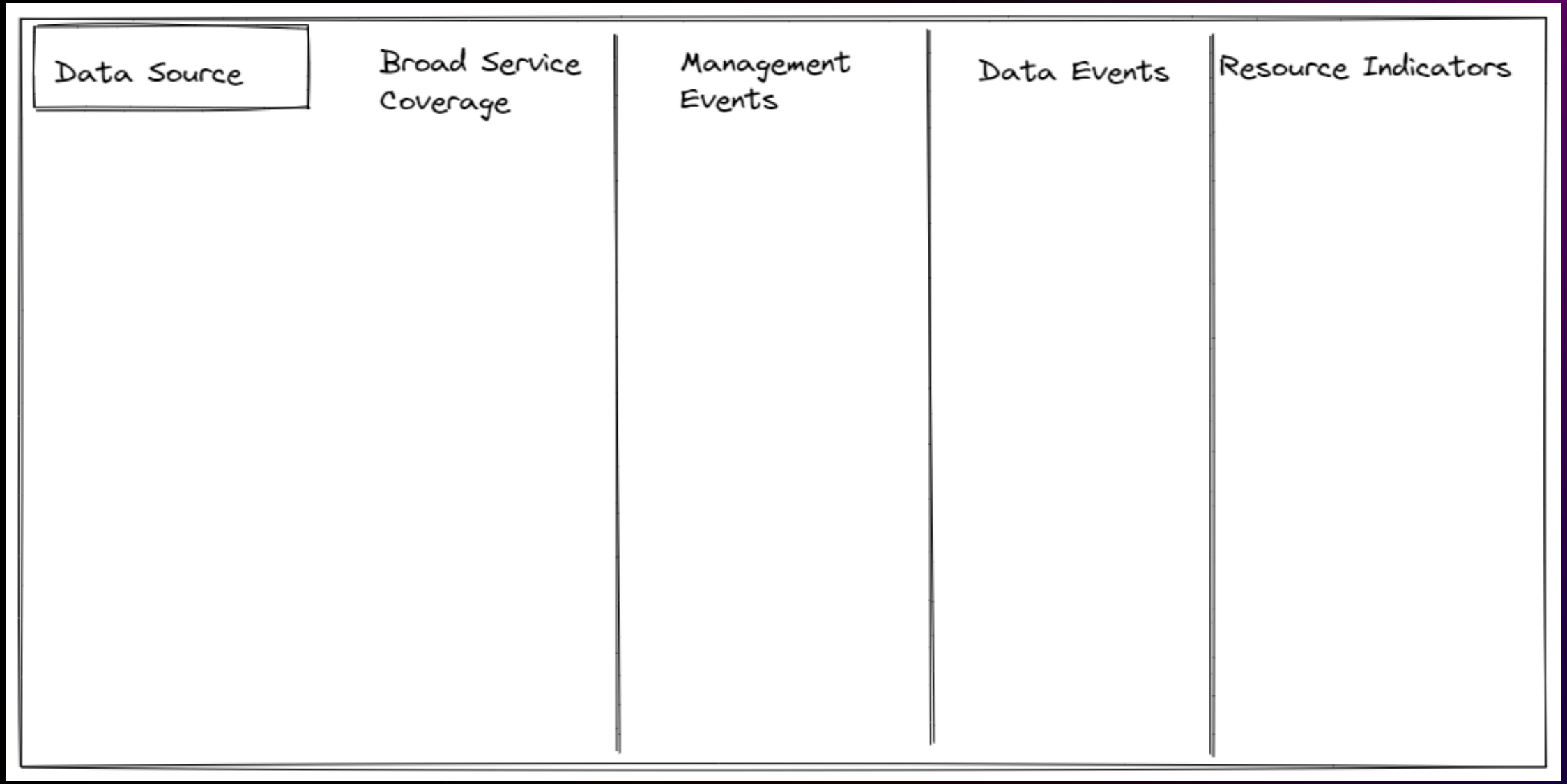
Application behavior and resource dependencies

Identity migration – account anchors



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Platform data



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Platform data

Data Source	Broad Service Coverage	Management Events	Data Events	Resource Indicators
CloudTrail	✓	✓	✗	✓

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Platform data

Data Source	Broad Service Coverage	Management Events	Data Events	Resource Indicators
CloudTrail	✓	✓	✗	✓
S3 Access Logs	✗	✗	✓	✓

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Platform data

Data Source	Broad Service Coverage	Management Events	Data Events	Resource Indicators
CloudTrail	✓	✓	✗	✓
S3 Access Logs	✗	✗	✓	✓
IAM Access Advisor	✓	✓	✓*	✗

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Platform data

Data Source	Broad Service Coverage	Management Events	Data Events	Resource Indicators
CloudTrail	✓	✓	✗	✓
S3 Access Logs	✗	✗	✓	✓
IAM Access Advisor	✓	✓	✓*	✗
IAM Policies	✓	*	*	*

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Client-side visibility



SDK visibility options

- Client-side metrics (AWS_CSM_ENABLED)
- Proxy (HTTP_PROXY)
- SDK – distribute patched version
- SDK – runtime-enabled intercepts

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Instrumentation

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some context on Netflix

Paved-path languages

- Majority Java: >75%
- Ubiquitous libraries for runtime, logs, security, etc.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some context on Netflix

Paved-path languages

- Majority Java: >75%
- Ubiquitous libraries for runtime, logs, security, etc.

Immutable infrastructure

- No changes to running apps
- Frequent re-bakes/re-deploys

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Some context on Netflix

Paved-path languages

- Majority Java: >75%
- Ubiquitous libraries for runtime, logs, security, etc.

Consolidated logging

- (Most) app logs shipped to central data store via gateway or agent
- Logs can be queried by app name and/or per-log-entry tags

Immutable infrastructure

- No changes to running apps
- Frequent re-bakes/re-deploys

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

The right hook

- In-process, data-rich
- No per-client/per-session code
- Transparent to application owners
- Safe enough for production code

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

The right hook

- In-process, data-rich
- No per-client/per-session code
- Transparent to application owners
- Safe enough for production code

What does AWS already provide? AWS X-Ray

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

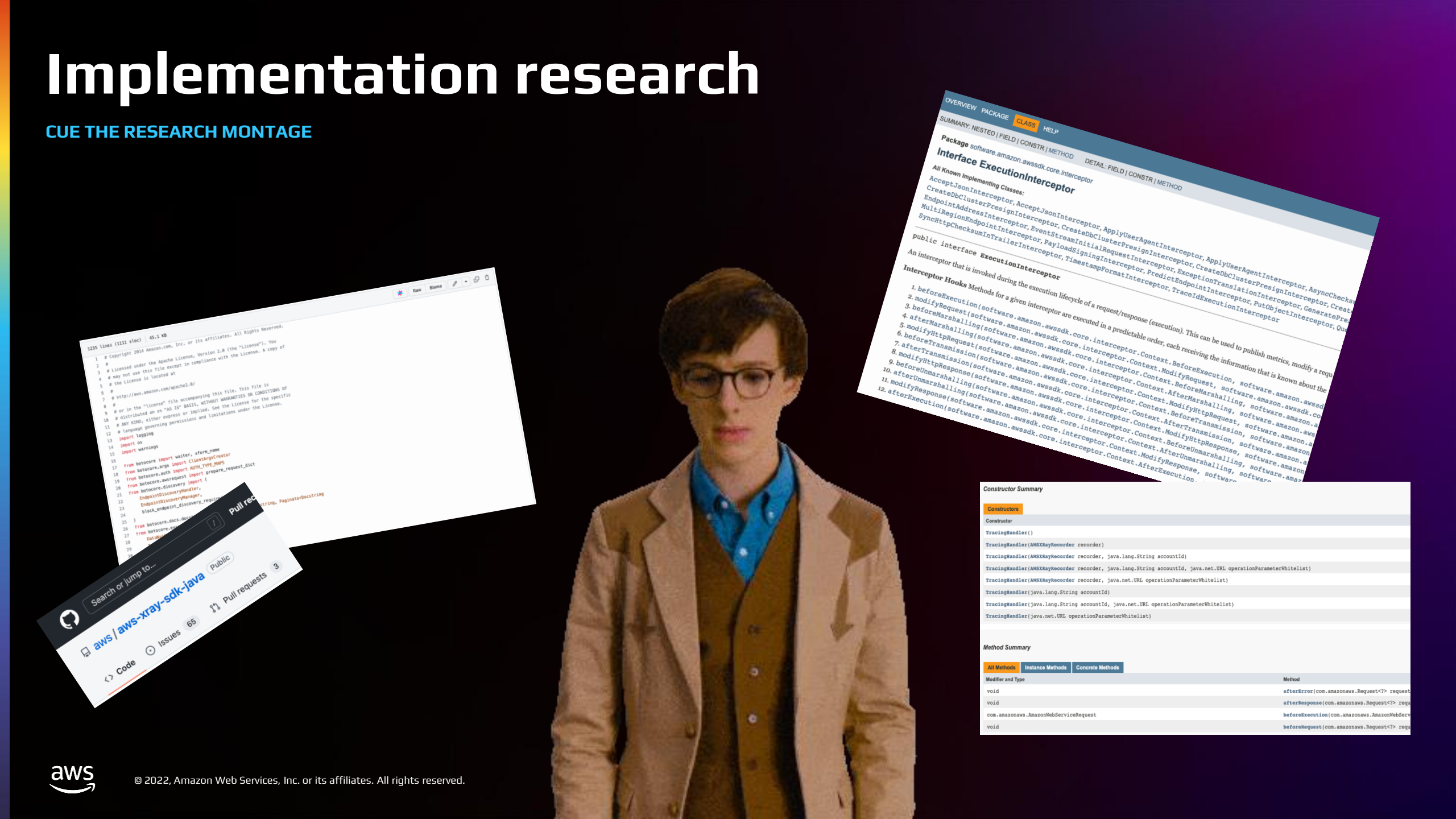
Just use AWS X-Ray?

- Not all apps follow a trace-friendly model
- Not all resource types are supported
- Expectation of code changes

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation research

CUE THE RESEARCH MONTAGE



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

Interceptor Registration Interceptors can be registered in one of many ways.

1. *Override Configuration Interceptors* are the most common method for SDK users to register an interceptor. These interceptors are explicitly added to the client builder's override configuration when a client is created using the `ClientOverrideConfiguration.Builder.addExecutionInterceptor(ExecutionInterceptor)` method.
2. *Global Interceptors* are interceptors loaded from the classpath for all clients. When any service client is created by a client builder, all jars on the classpath (from the perspective of the current thread's classloader) are checked for a file named '/software/amazon/awssdk/global/handlers/execution.interceptors'. Any interceptors listed in these files (new line separated) are instantiated using their default constructor and loaded into the client.
3. *Service Interceptors* are interceptors loaded from the classpath for a particular service's clients. When a service client is created by a client builder, all jars on the classpath (from the perspective of the current thread's classloader) are checked for a file named '/software/amazon/awssdk/services/{service}/execution.interceptors', where {service} is the package name of the service client. Any interceptors listed in these files (new line separated) are instantiated using their default constructor and loaded into the client.

https://sdk.amazonaws.com/java/api/latest/software/amazon/awssdk/core/interceptor/ExecutionInterceptor.html

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

Java

resources/software/amazon/awssdk/global/handlers/execution.interceptors

```
1 com.netflix.exampleinstrumentor.Instrumentor
```

src/main/java/com/netflix/exampleinstrumentor/Instrumentor.java

```java
package com.netflix.exampleinstrumentor;

import software.amazon.awssdk.core.interceptor.ExecutionInterceptor;

...

/**
 *
 * Instrumentor implementation for AWS SDK for Java V2
 *
 */
public class Instrumentor extends ExecutionInterceptor {

  ...

  public void beforeExecution(
    Context.BeforeExecution context,
    ExecutionAttributes executionAttributes
  ) {
    process(context, executionAttributes);
  }
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

Java

resources/software/amazon/awssdk/global/handlers/execution.interceptors

```
com.netflix.exampleinstrumentor.Instrumentor
```



src/main/java/com/netflix/exampleinstrumentor/Instrumentor.java

```java
package com.netflix.exampleinstrumentor;

import software.amazon.awssdk.core.interceptor.ExecutionInterceptor;

...

/**
 *
 * Instrumentor implementation for AWS SDK for Java V2
 *
 */
public class Instrumentor extends ExecutionInterceptor {

   ...

   public void beforeExecution(
      Context.BeforeExecution context,
      ExecutionAttributes executionAttributes
   ) {
      process(context, executionAttributes);
   }
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

```
879
880     def _make_api_call(self, operation_name, api_params):
881         operation_model = self._service_model.operation_model(operation_name)
882         service_name = self._service_model.service_name
883         history_recorder.record(
884             'API_CALL',
885             {
886                 'service': service_name,
887                 'operation': operation_name,
888                 'params': api_params,
889             },
890         )
```

https://github.com/boto/botocore/blob/develop/botocore/client.py

```
1  import logging
2
3  HISTORY_RECORDER = None
4  logger = logging.getLogger(__name__)
5
6
7  class BaseHistoryHandler:
8      def emit(self, event_type, payload, source):
9          raise NotImplementedError('emit()')
10
11
12 class HistoryRecorder:
13     def __init__(self):
14         self._enabled = False
15         self._handlers = []
16
17     def enable(self):
18         self._enabled = True
19
20     def disable(self):
21         self._enabled = False
22
23     def add_handler(self, handler):
24         self._handlers.append(handler)
25
26     def record(self, event_type, payload, source='BOTOCORE'):
27         if self._enabled and self._handlers:
28             for handler in self._handlers:
29                 try:
30                     handler.emit(event_type, payload, source)
31                 except Exception:
32                     # Never let the process die because we had a failure in
33                     # a record collection handler.
34                     logger.debug(
35                         "Exception raised in %s.", handler, exc_info=True
36                     )
37
38
39 def get_global_history_recorder():
40     global HISTORY_RECORDER
41     if HISTORY_RECORDER is None:
42         HISTORY_RECORDER = HistoryRecorder()
43     return HISTORY_RECORDER
```

https://github.com/boto/botocore/blob/develop/botocore/history.py

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

```python
from botocore.history import BaseHistoryHandler, get_global_history_recorder

class ExampleHistoryHandler(BaseHistoryHandler):

    def __init__(self):
        # register ourselves as a handler
        HISTORY_RECORDER = get_global_history_recorder()
        HISTORY_RECORDER.add_handler(self)
        HISTORY_RECORDER.enable()

    def emit(self, event_type, payload, source):
        # do stuff
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

```python
from botocore.history import BaseHistoryHandler, get_global_history_recorder

class ExampleHistoryHandler(BaseHistoryHandler):

    def __init__(self):
        # register ourselves as a handler
        HISTORY_RECORDER = get_global_history_recorder()
        HISTORY_RECORDER.add_handler(self)
        HISTORY_RECORDER.enable()

    def emit(self, event_type, payload, source):
        # do stuff
```

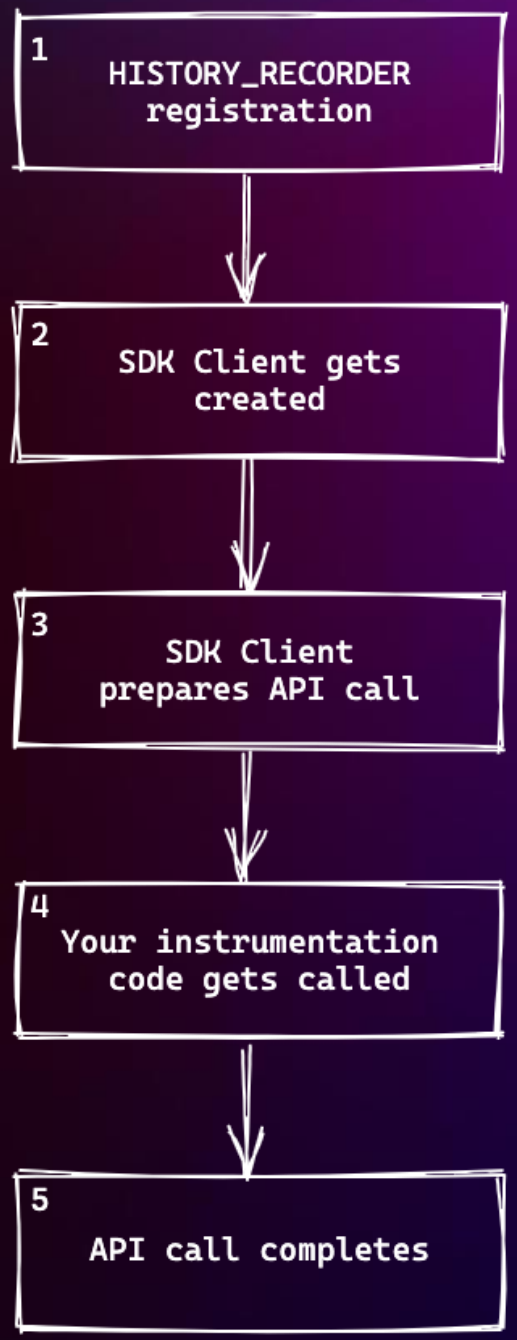


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Example: Intercepting an SQS:SendMessage call

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

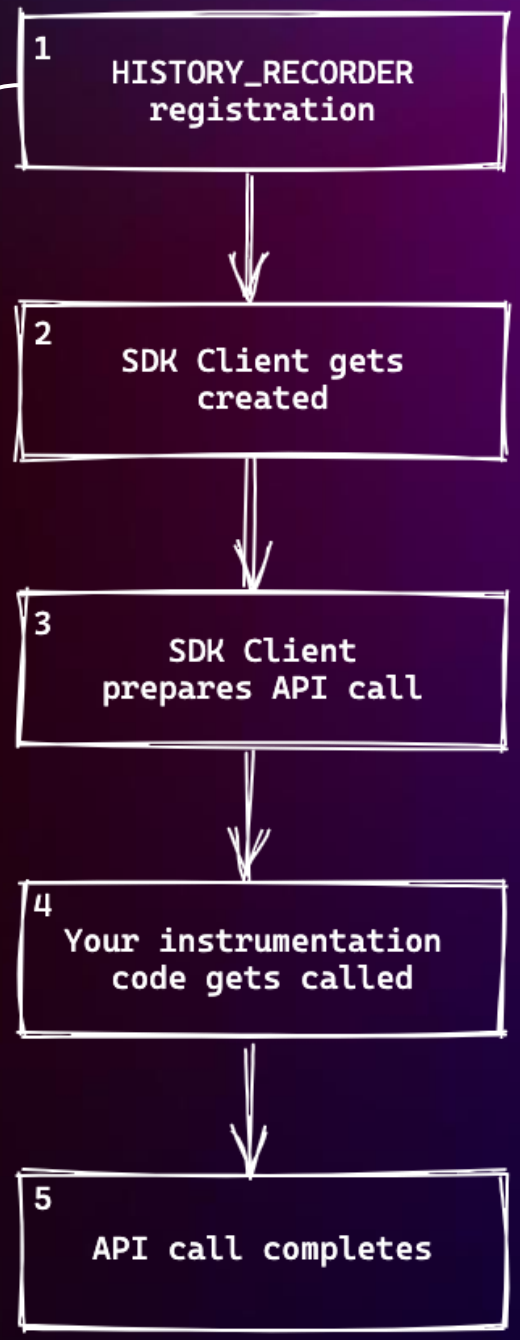


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

(in instrumentation)

```
6         # register ourselves as a handler
7         HISTORY_RECORDER = get_global_history_recorder()
8         HISTORY_RECORDER.add_handler(self)
9         HISTORY_RECORDER.enable()
10
```

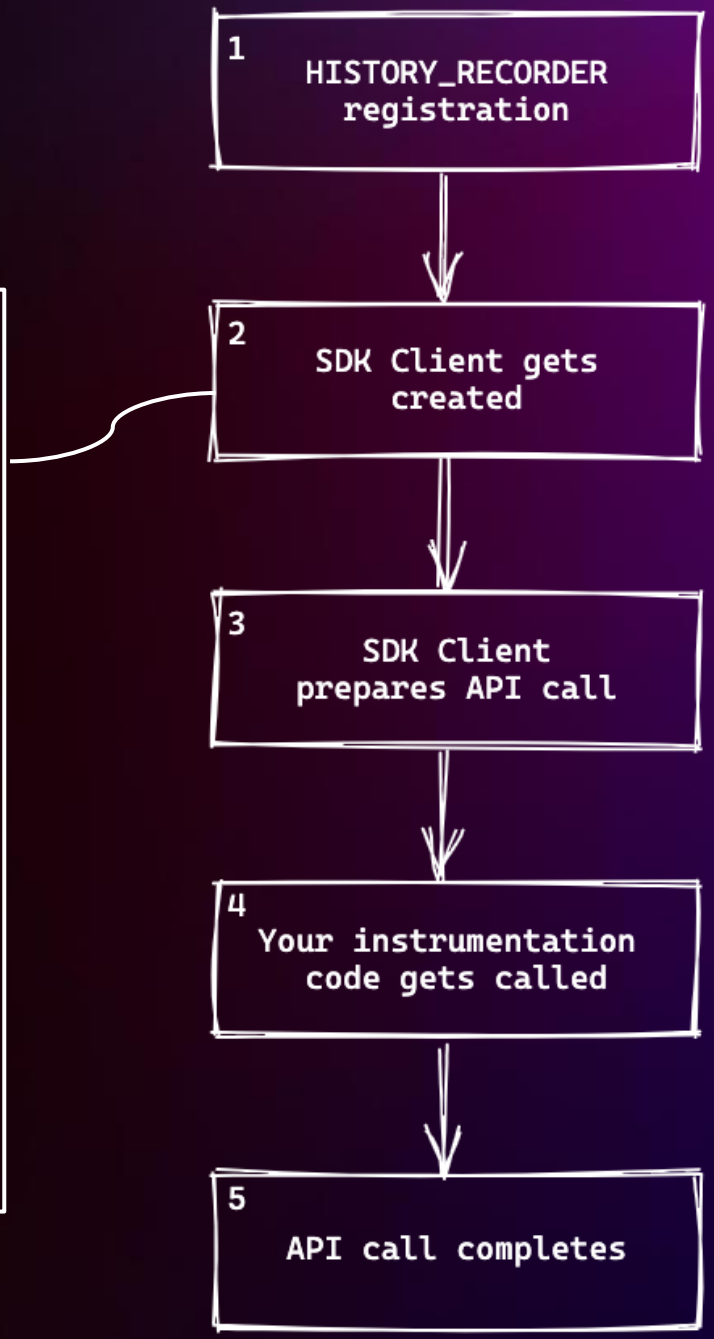


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

(in application)

```
1 import boto3
2
3 client = boto3.client('sqs')
```

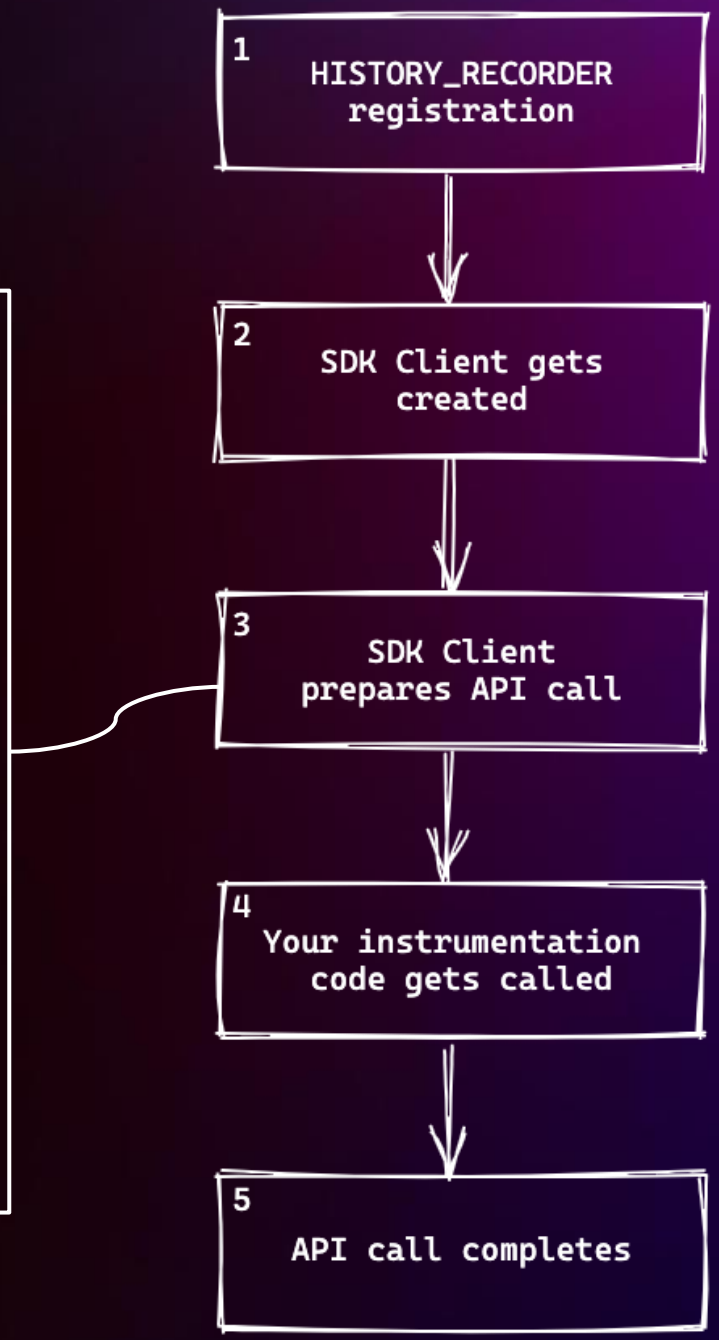


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

(in application)

```
5 client.send_message(
6     QueueUrl='https://sqs.us-east-1.amazonaws.com/111111111111/example-queue',
7     MessageBody='an example message',
8 )
```

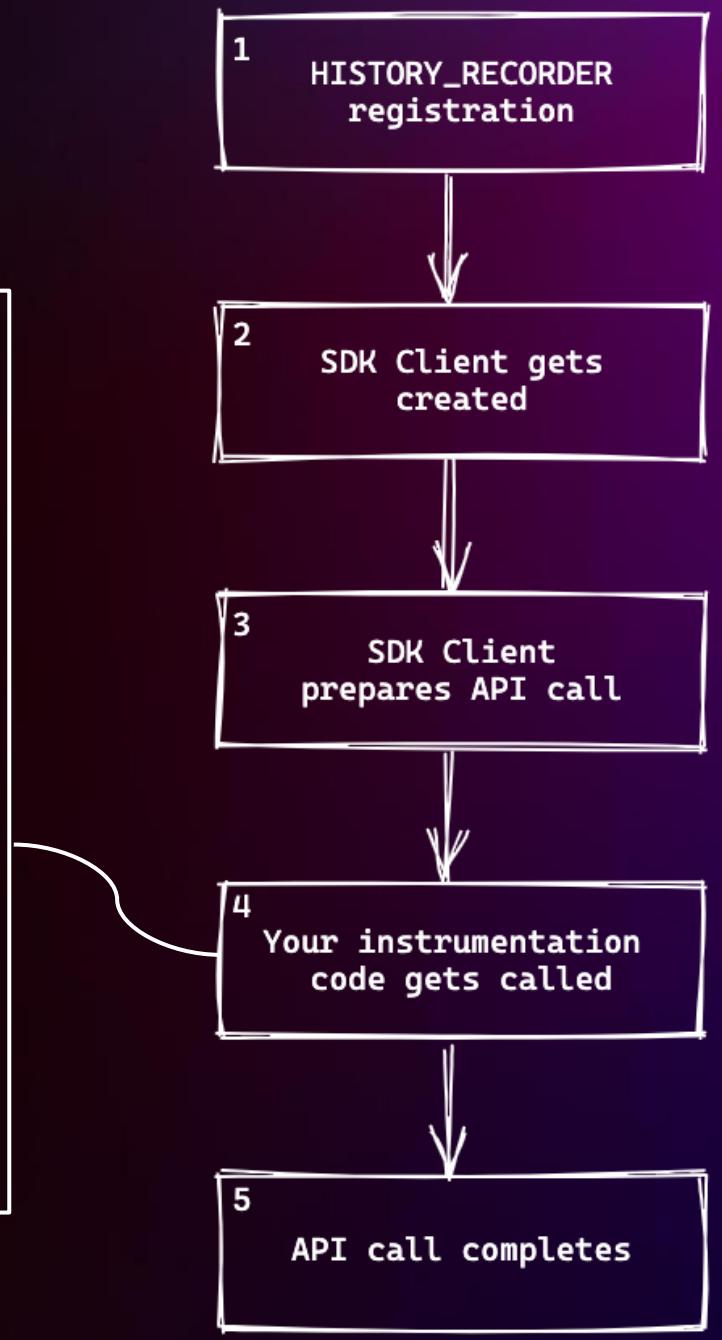


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

```
11      def emit(self, event_type, payload, source):
12          # do stuff
13          print('event_type =', event_type)
14          print('payload =', payload)
15          print('source =', source)
```

(in instrumentation)



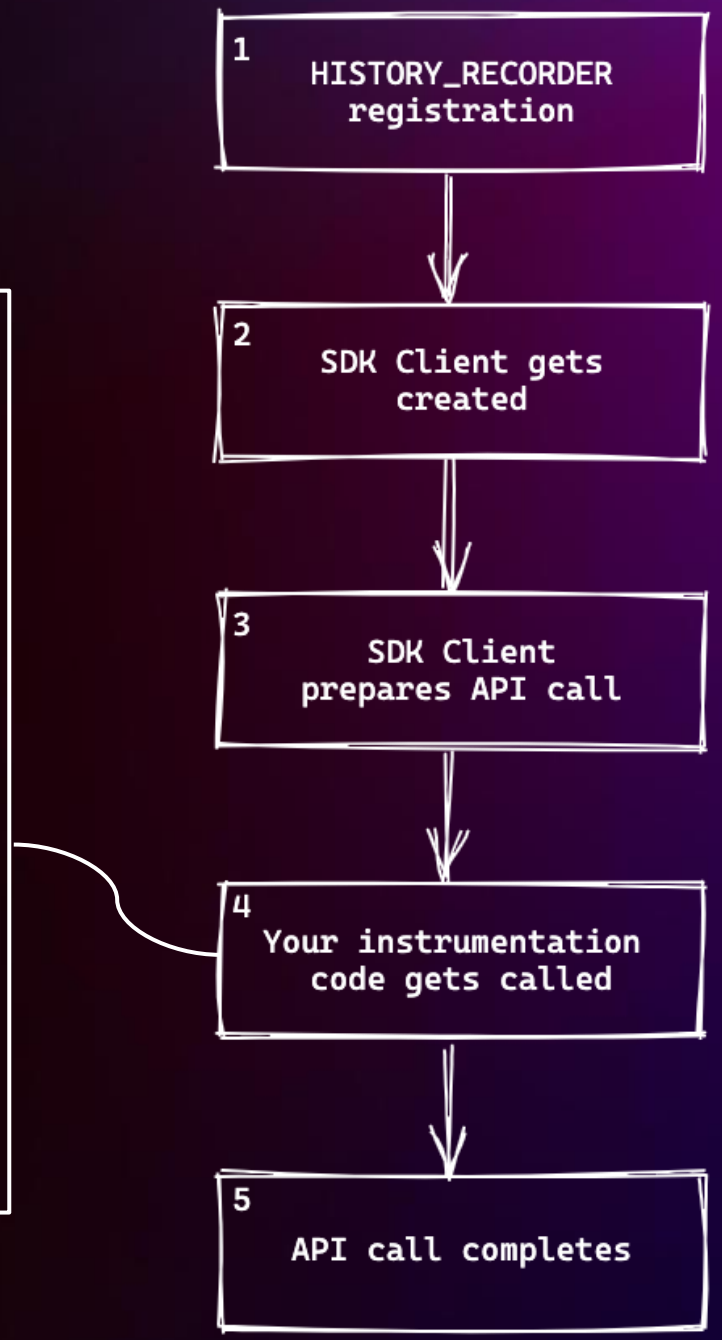
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept

```
11      def emit(self, event_type, payload, source):
12          # do stuff
13          print('event_type =', event_type)
14          print('payload =', payload)
15          print('source =', source)
```

(in instrumentation)

```
event_type = API_CALL
payload = {'service': 'sqs', 'operation': 'SendMessage', 'params': {'QueueUrl': 'https://sqs.us-east-1.
amazonaws.com/111111111111/example-queue', 'MessageBody': 'an example message'}}
source = BOTOCORE
```



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

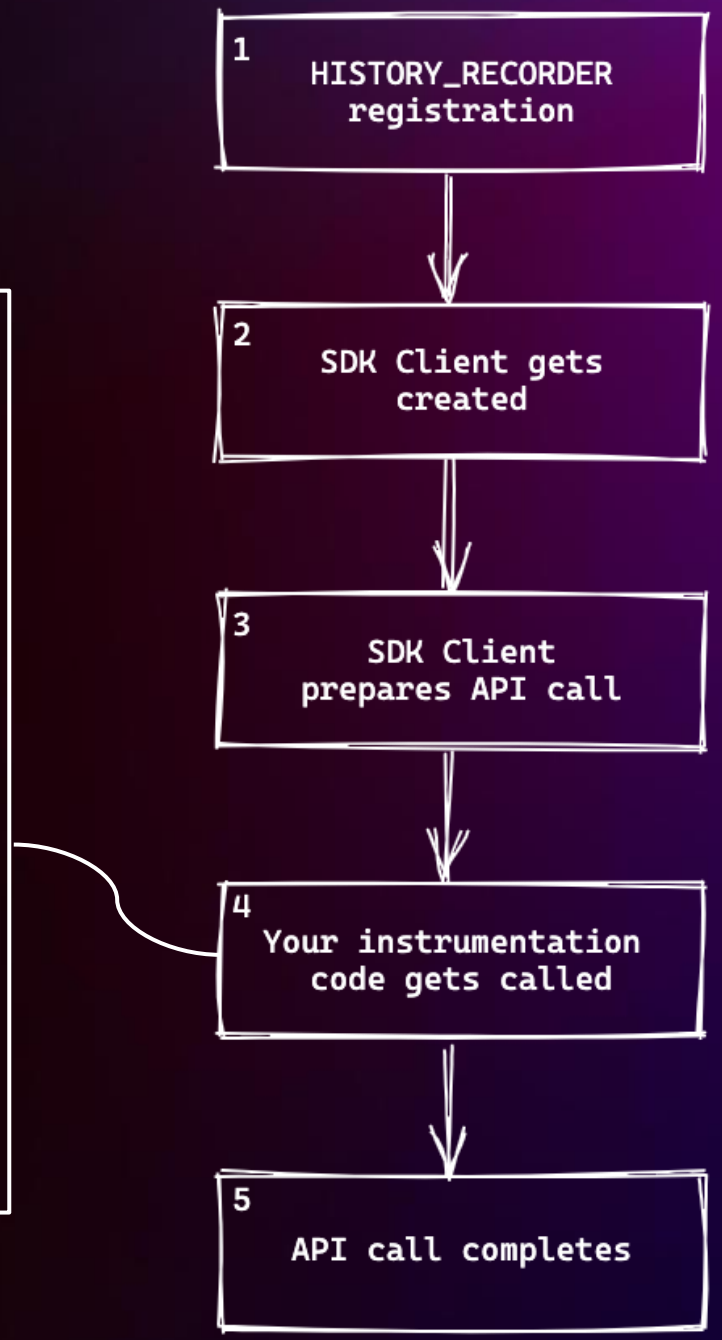
Implementation intercept

```
11      def emit(self, event_type, payload, source):
12          # do stuff
13          print('event_type =', event_type)
14          print('payload =', payload)
15          print('source =', source)
```

(in instrumentation)

```
event_type = API_CALL
payload = {'service': 'sqs', 'operation': 'SendMessage', 'params': {'QueueUrl': 'https://sqs.us-east-1.
amazonaws.com/111111111111/example-queue', 'MessageBody': 'an example message'}}
source = BOTOCORE
```

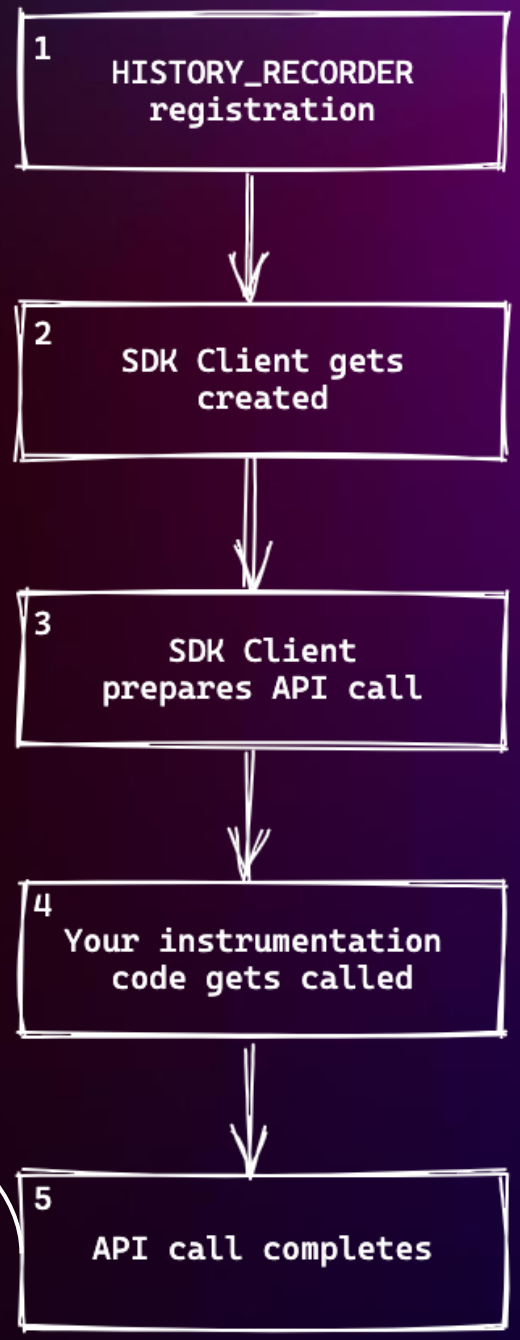
```
{
  "operation": "SendMessage",
  "params": {
    "MessageBody": "an example message",
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/111111111111/example-queue"
  },
  "service": "sqs"
}
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation intercept





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation deployment

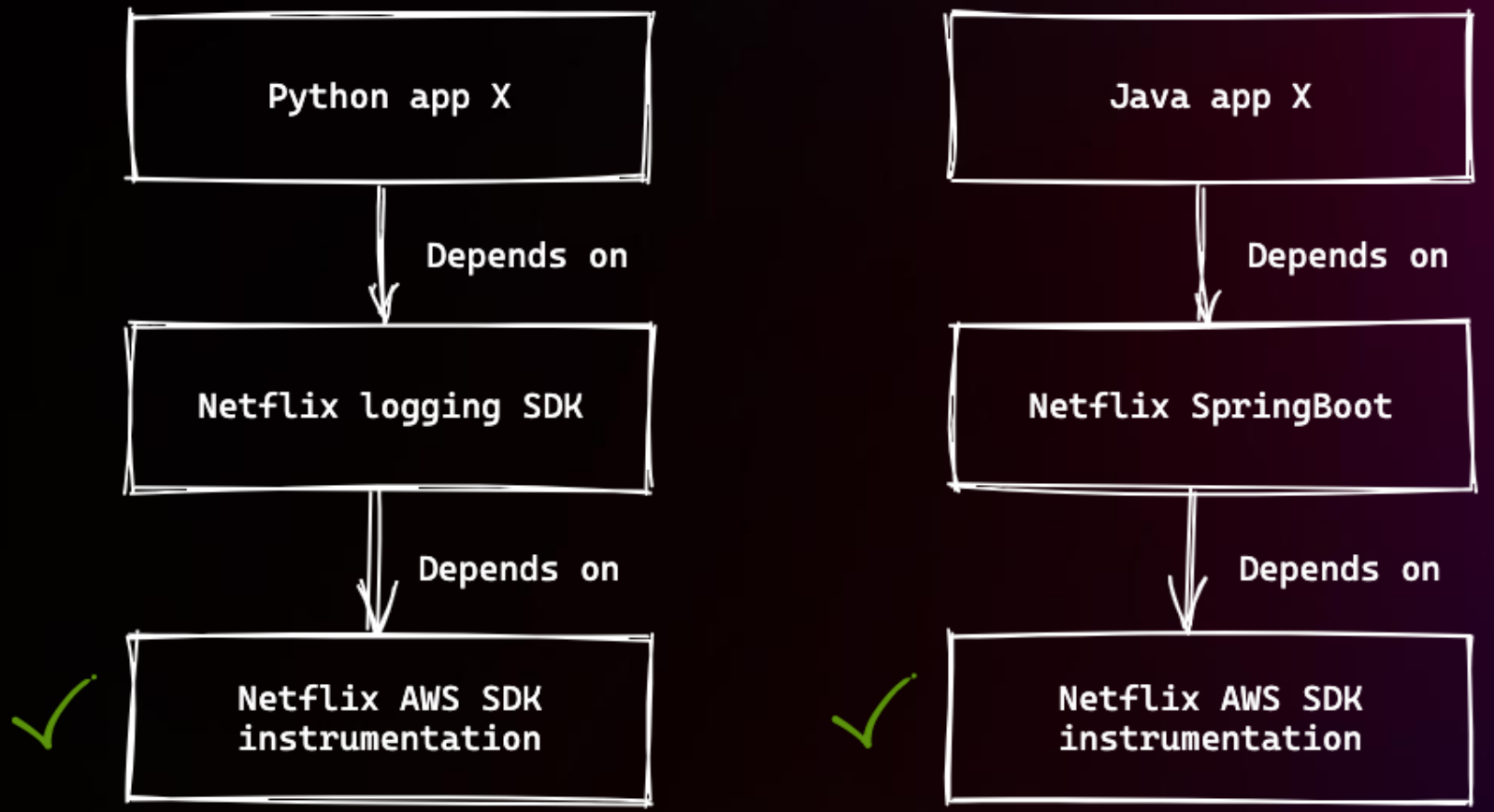
Getting our code in

- **Task**: adding a library dependency to each application
- Our approach: add our libraries as dependencies of “baseline” packages: our organization’s logging and runtime
- Still much easier than maintaining + enforcing a fork of AWS SDKs

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation deployment



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Post-capture

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Managing data sensitivity and volume

```
payload = {'service': 'sqs', 'operation': 'SendMessage', 'params': {'QueueUrl':
'https://sqs.us-east-1.amazonaws.com/111111111111/example-queue', 'MessageBody': 'an example message'}}
payload = {'service': 'sqs', 'operation': 'SendMessage', 'params': {'QueueUrl':
'https://sqs.us-east-1.amazonaws.com/111111111111/example-queue', 'MessageBody': 'another message'}}
payload = {'service': 'sqs', 'operation': 'SendMessage', 'params': {'QueueUrl':
'https://sqs.us-east-1.amazonaws.com/111111111111/example-queue', 'MessageBody': 'and then another'}}
```




```
payload = {'service': 'sqs', 'operation': 'SendMessage', 'params': {'QueueUrl':
'https://sqs.us-east-1.amazonaws.com/111111111111/example-queue'}
```

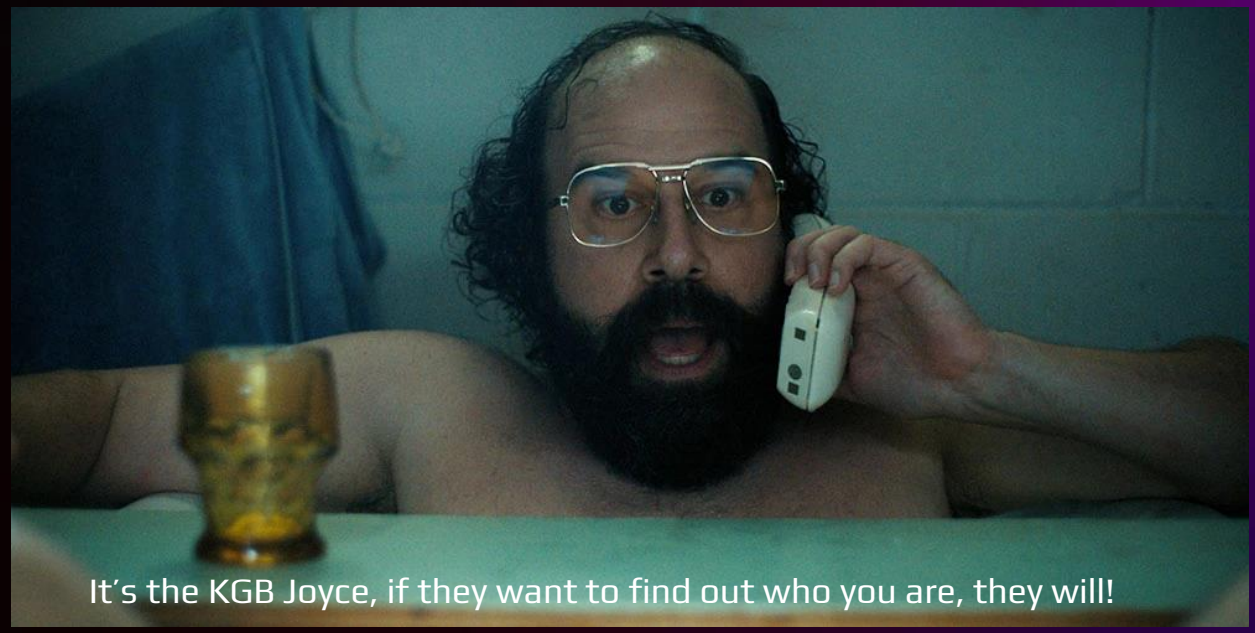
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://github.com/aws/aws-xray-sdk-java/blob/master/aws-xray-recorder-sdk-aws-sdk/src/main/resources/com/amazonaws/xray/handlers/DefaultOperationParameterWhitelist.json

Identifying callers

Who is the caller?

- Init
  - SDK credential resolution
- Signing time
  - Credential intercept
- Omit sts.get_caller_identity from inspection



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data export

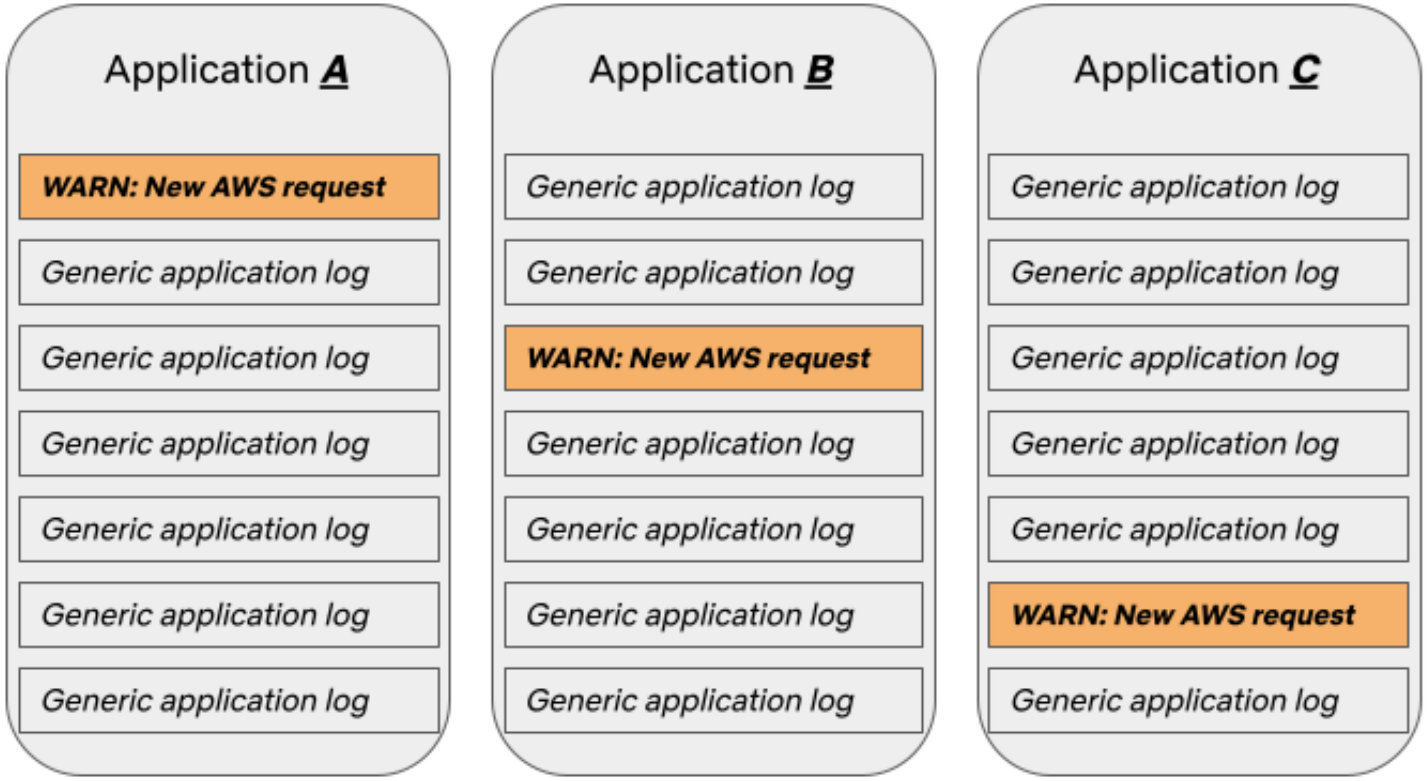
Getting our data out

- Language merge
  - JSON message spec
- Limited only by what we can do in code
- In-line with API calls; lightweight is best
- Our approach: adding to the application log stream with metadata (instance-local logging daemon)

aws

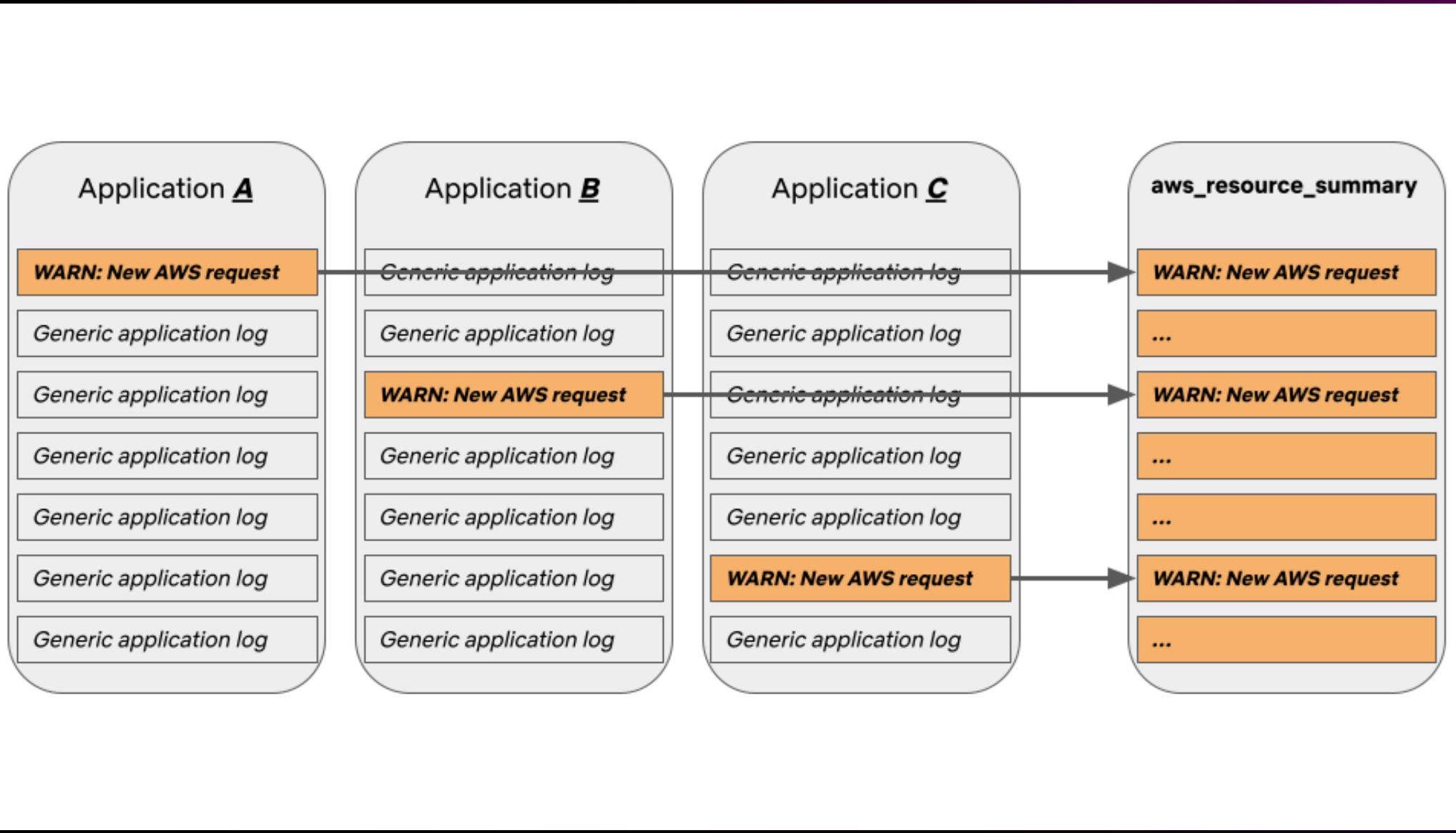
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Off-instance aggregation



aws ©

Off-instance aggregation



aws
©

Off-instance processing

- Scheduled job reformats log events

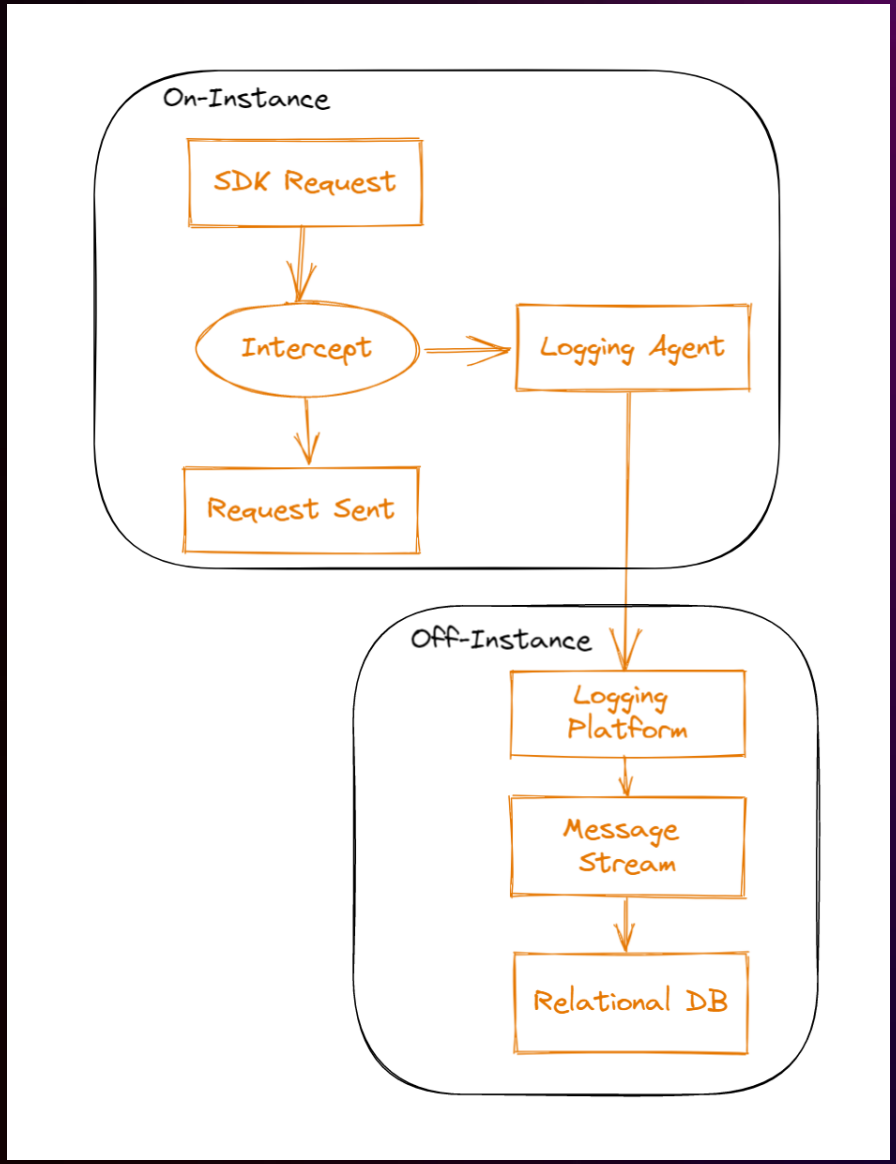
  QueueUrl=https://sqs.us-east...../example-queue

- To...

  Key=QueueUrl

  Value=https://sqs.us-east1..../example-queue

- Submits to DB via Stream



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Answering the question

What was the question again? Oh right.

Which AWS resources does a given application interact with?

```
SELECT DISTINCT app, service, operation, key, value

WHERE app = 'bigbrother';
```

app	ser...	operation	key ↑	value
bigbrother	s3	GetObject	Bucket	-cloudtrail
bigbrother	s3	GetObject	Bucket	-cloudtrail-bucket
bigbrother	s3	GetObject	Bucket	-prod-us-west-2
bigbrother	sqs	DeleteMessage	QueueUrl	arn:aws:sqs:us-east-1: cloudtrail-event-queue
bigbrother	sqs	ReceiveMessage	QueueUrl	arn:aws:sqs:us-east-1: _trail_queue
bigbrother	sqs	DeleteMessage	QueueUrl	arn:aws:sqs:us-east-1: _trail_queue
bigbrother	sqs	ReceiveMessage	QueueUrl	arn:aws:sqs:us-east-1: -cloudtrail-event-queue

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In closing...

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lessons learned

Technical

- Parameter name overlap
- High resource-touch applications
- SDKs have very different views of AWS
  - Intercept methods, Objects vs. Dicts, Request Lifecycle

Strategic

- Piggyback on a “carrot”

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What’s next/What’s possible

- Using usage data to rewrite resource policies to allow migration of application identities
- Inspecting responses to identify permissions failures in real time
- Understanding full scope of operational dependencies for apps

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Questions?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Nick Siow
nsiow@netflix.com

Scott Pack
scottpack@netflix.com

Prateek Sharma



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.