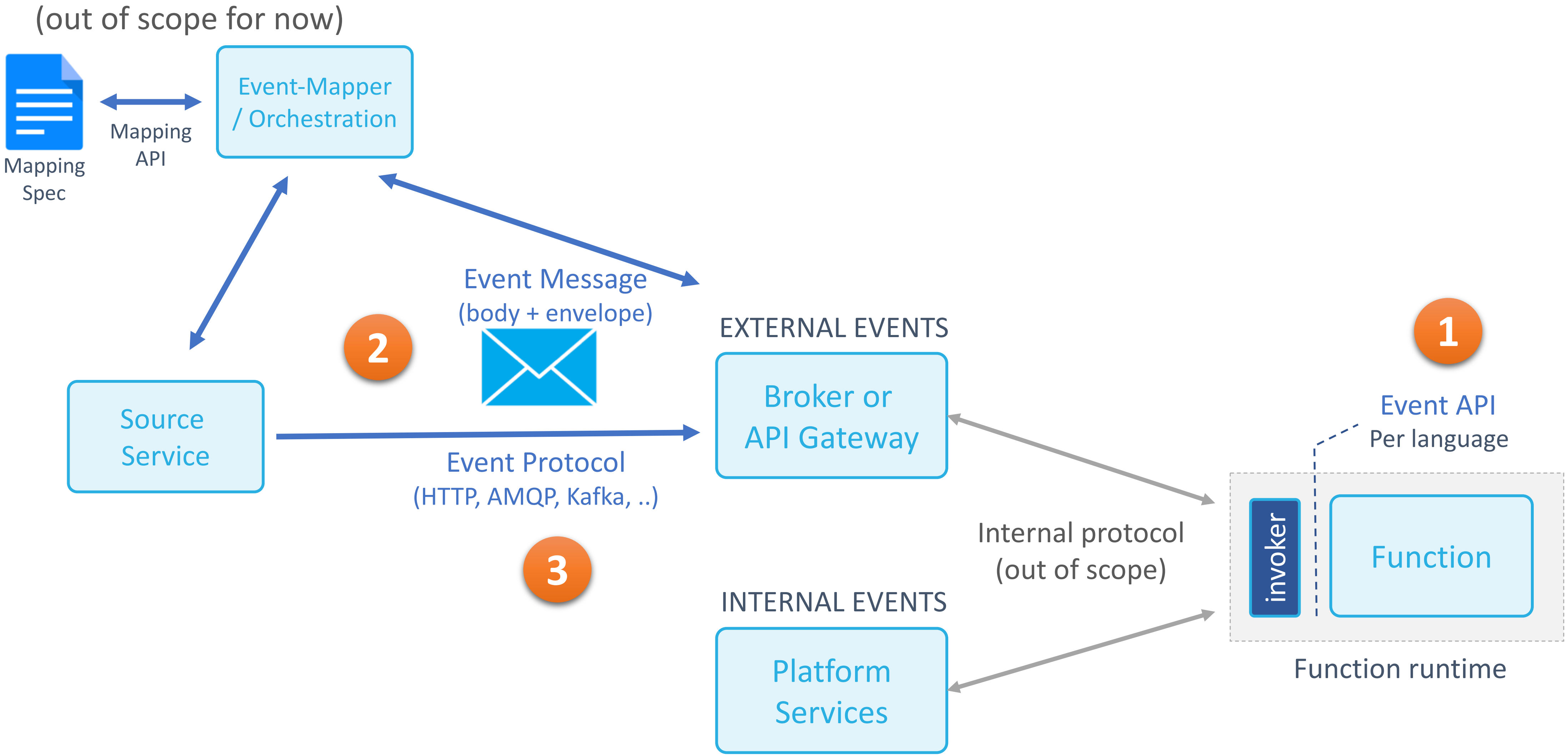
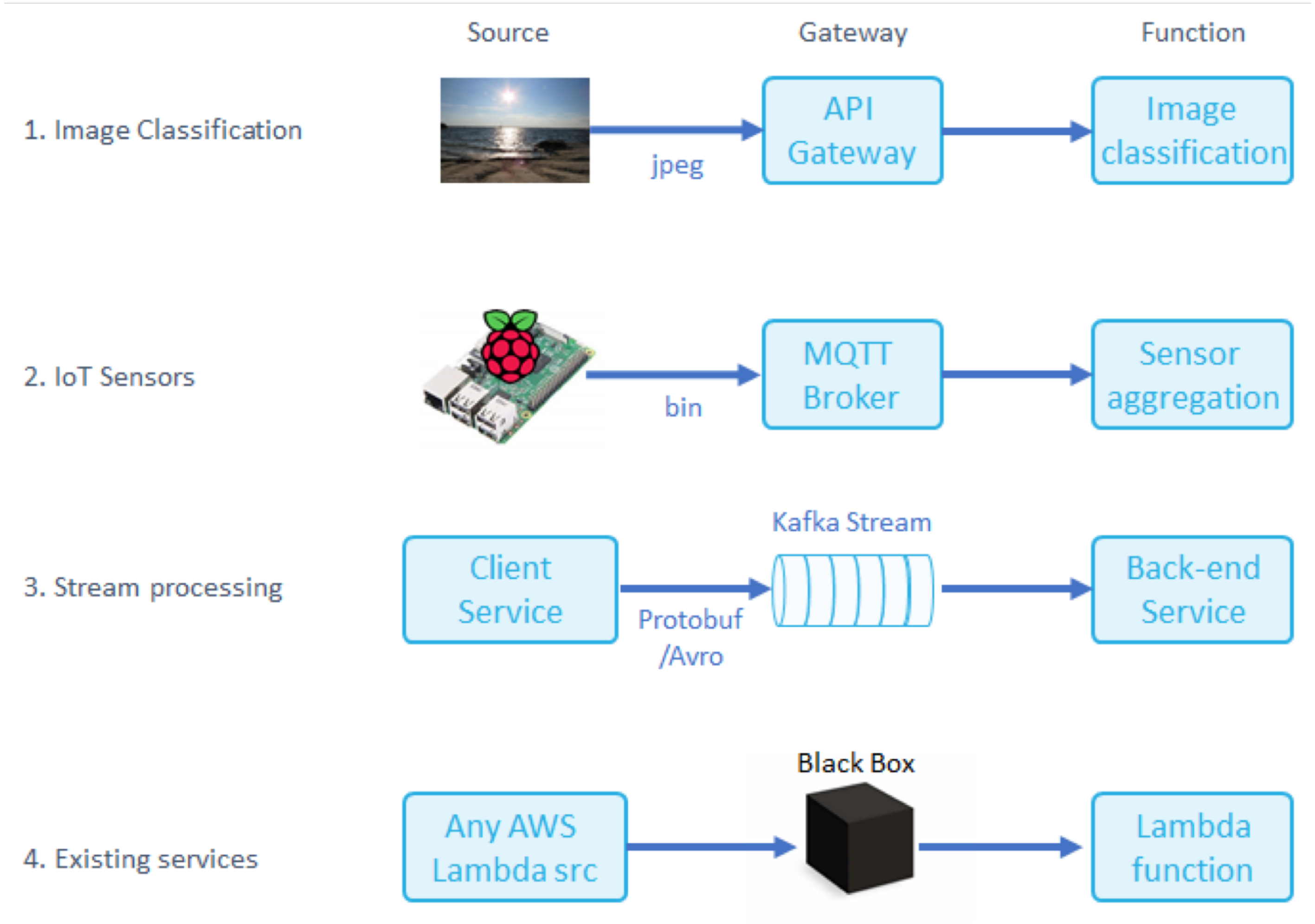
Cloud Events Problem Scope and Relations



Determining Your Event Metadata Today

```
function getLambdaEventSource(event) {
   if (event.Records && event.Records[0].cf) return 'isCloudfront';
   if (event.configRuleId && event.configRuleName && event.configRuleArn) return 'isAwsConfig';
   if (event.Records && (event.Records[0].eventSource === 'aws:codecommit')) return 'isCodeCommit';
   if (event.authorizationToken === "incoming-client-token") return 'isApiGatewayAuthorizer';
   if (event.StackId && event.RequestType && event.ResourceType) return 'isCloudFormation';
   if (event.Records && (event.Records[0].eventSource === 'aws:ses')) return 'isSes';
   if (event.pathParameters && event.pathParameters.proxy) return 'isApiGatewayAwsProxy';
   if (event.source === 'aws.events') return 'isScheduledEvent';
   if (event.awslogs && event.awslogs.data) return 'isCloudWatchLogs';
   if (event.Records && (event.Records[0].EventSource === 'aws:sns')) return 'isSns';
   if (event.Records && (event.Records[0].eventSource === 'aws:dynamodb')) return 'isDynamoDb';
   if (event.records && event.records[0].approximateArrivalTimestamp) return 'isKinesisFirehose';
   if (event.records && event.deliveryStreamArn && event.deliveryStreamArn.startsWith('arn:aws:kinesis:'))
return 'isKinesisFirehose';
   if (event.eventType === 'SyncTrigger' && event.identityId && event.identityPoolId) return
'isCognitoSyncTrigger';
   if (event.Records && event.Records[0].eventSource === 'aws:kinesis') return 'isKinesis';
   if (event.Records && event.Records[0].eventSource === 'aws:s3') return 'isS3';
   if (event.operation && event.message) return 'isMobileBackend';
}
```

Events May Go Beyond Text & JSON



Source

Gateway

Function

1. Image Classification

jpeg

API Gateway

Image classification

2. IoT Sensors

bin

MQTT Broker

Sensor aggregation

3. Stream processing

Client Service

Protobuf /Avro

Kafka Stream

Back-end Service

4. Existing services

Any AWS Lambda src

Black Box

Lambda function

```go
type Event interface {
    // GetID returns the unique ID of the event
    GetID() ID
    // GetTriggerInfo retruns a trigger info provider (stuff from the event listener, protocol specific)
    GetTriggerInfo() TriggerInfoProvider
    // GetSchema returns the Event Body Schema
    GetSchema() string
    // GetContentType returns the content type of the body (data blob)
    GetContentType() string
    // GetBody returns the body (data blob) of the event
    GetBody() []byte
    // GetHeader returns the header (context attributes) by name
    GetHeader(string) string
    // GetHeaders loads all headers (context attributes) into a map of string
    GetHeaders() map[string]string
    // GetField returns the field (within the data/body) by name/path as an interface{}
    GetField(string) interface{}
   // GetFields loads all data/body fields into a map of string / interface{}
    GetFields() map[string]interface{}
    // GetTimestamp returns when the event originated
    GetTimestamp() time.Time

    …
}
```

Example Event API (Go)

Event API provide a way to pass the event envelope metadata (Schema, type, source, ..) and the event message data/body

The API should not expose any event/schema specific information, the event data is accessed via the data/body or the de-serialized version of the data (e.g. using Fields method)

Envelop attributes are provided as a map of strings to allow extensibility / optional data