



Simple Workflow

Distributed Out of the Box!

Serhiy Batyuk
JEEConf
May 21, 2016

Tech Lead at



LOHIKA

AWS SWF > **1 year**

Agenda

**SWF
Overview**

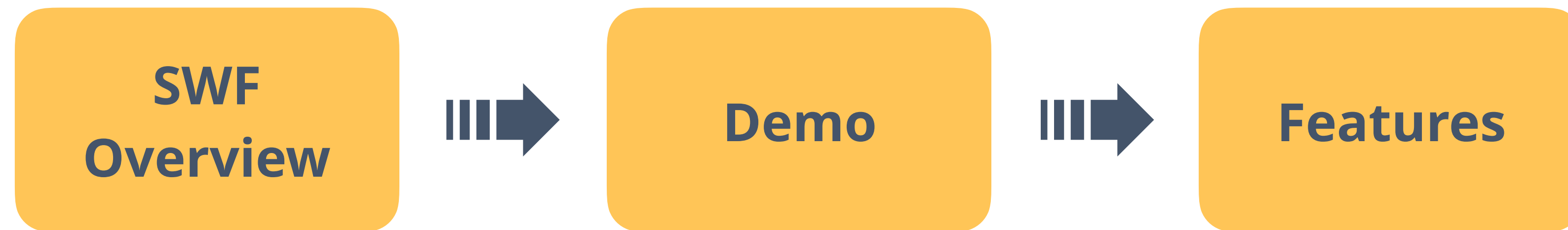
Agenda

**SWF
Overview**

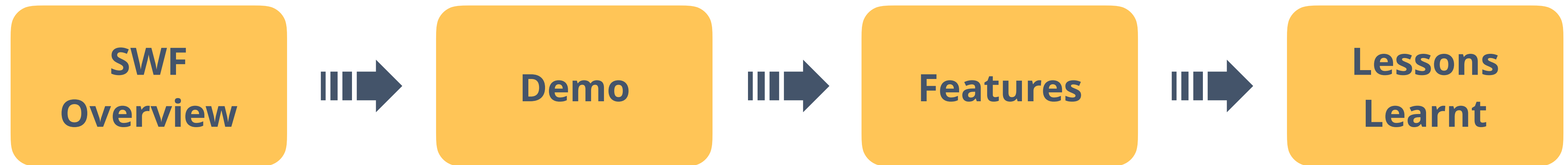


Demo

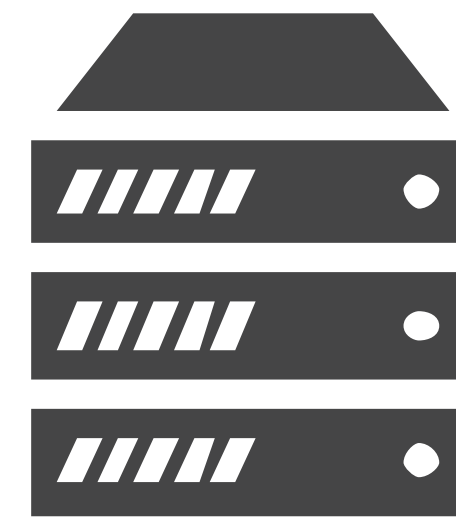
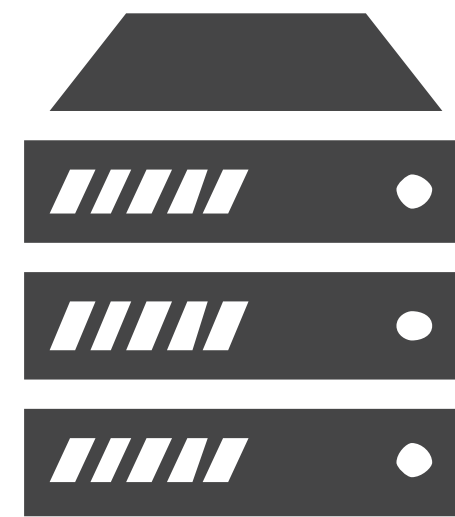
Agenda



Agenda



What Is **AWS Simple Workflow**?



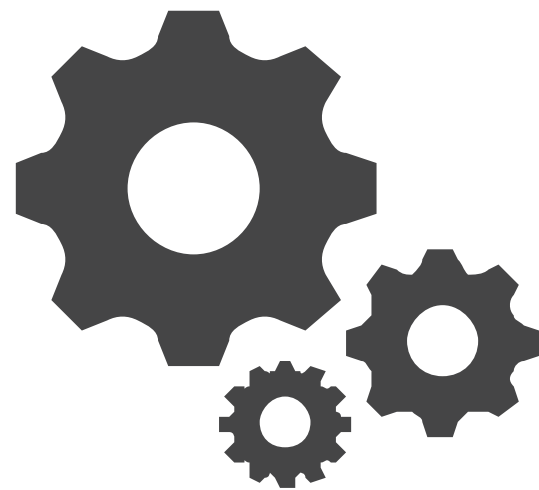
AWS SWF Use Cases



Media Processing



Infrastructure Provisioning



Data Processing



Report Generation



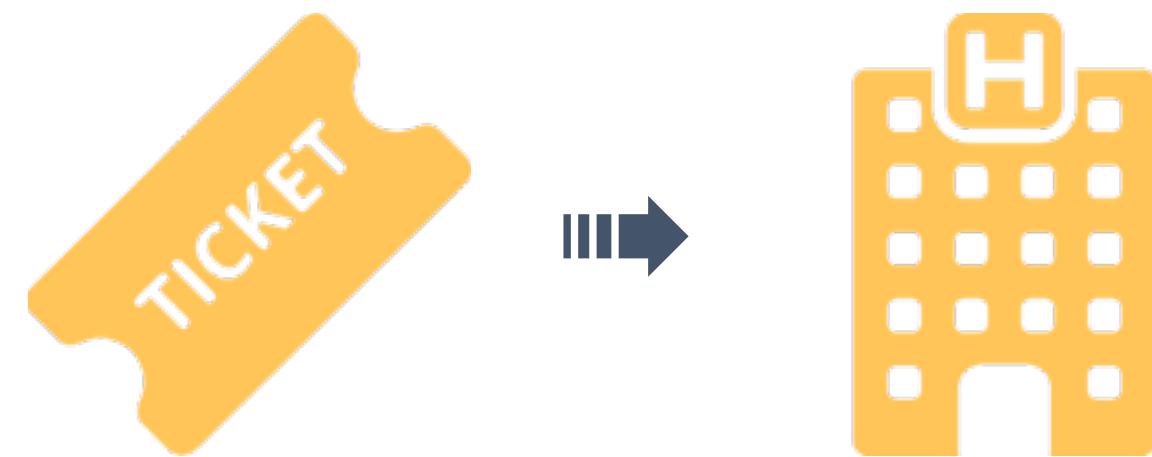
Business Processes



Preparing to Attend **JEEConf**



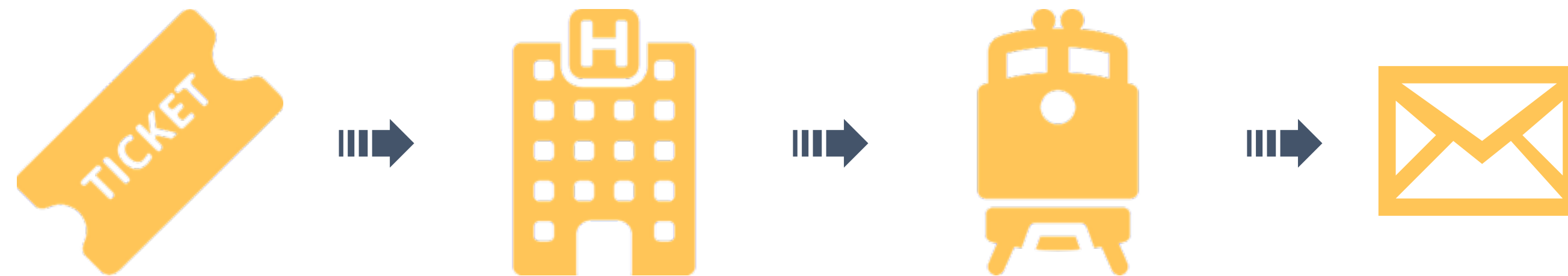
Preparing to Attend **JEEConf**



Preparing to Attend **JEEConf**

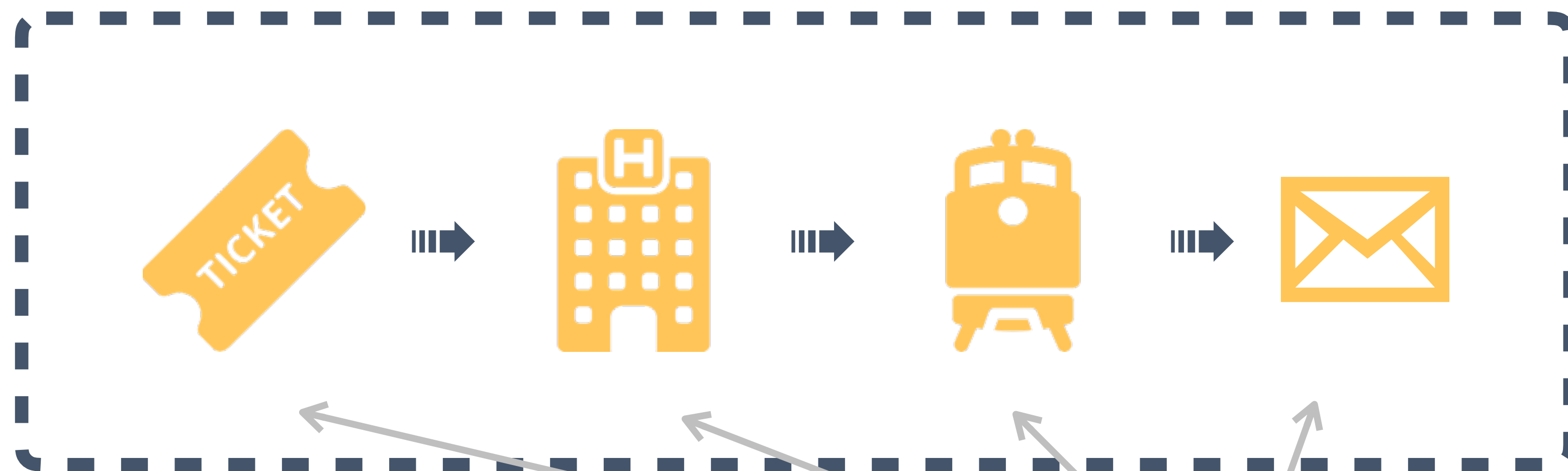


Preparing to Attend **JEEConf**



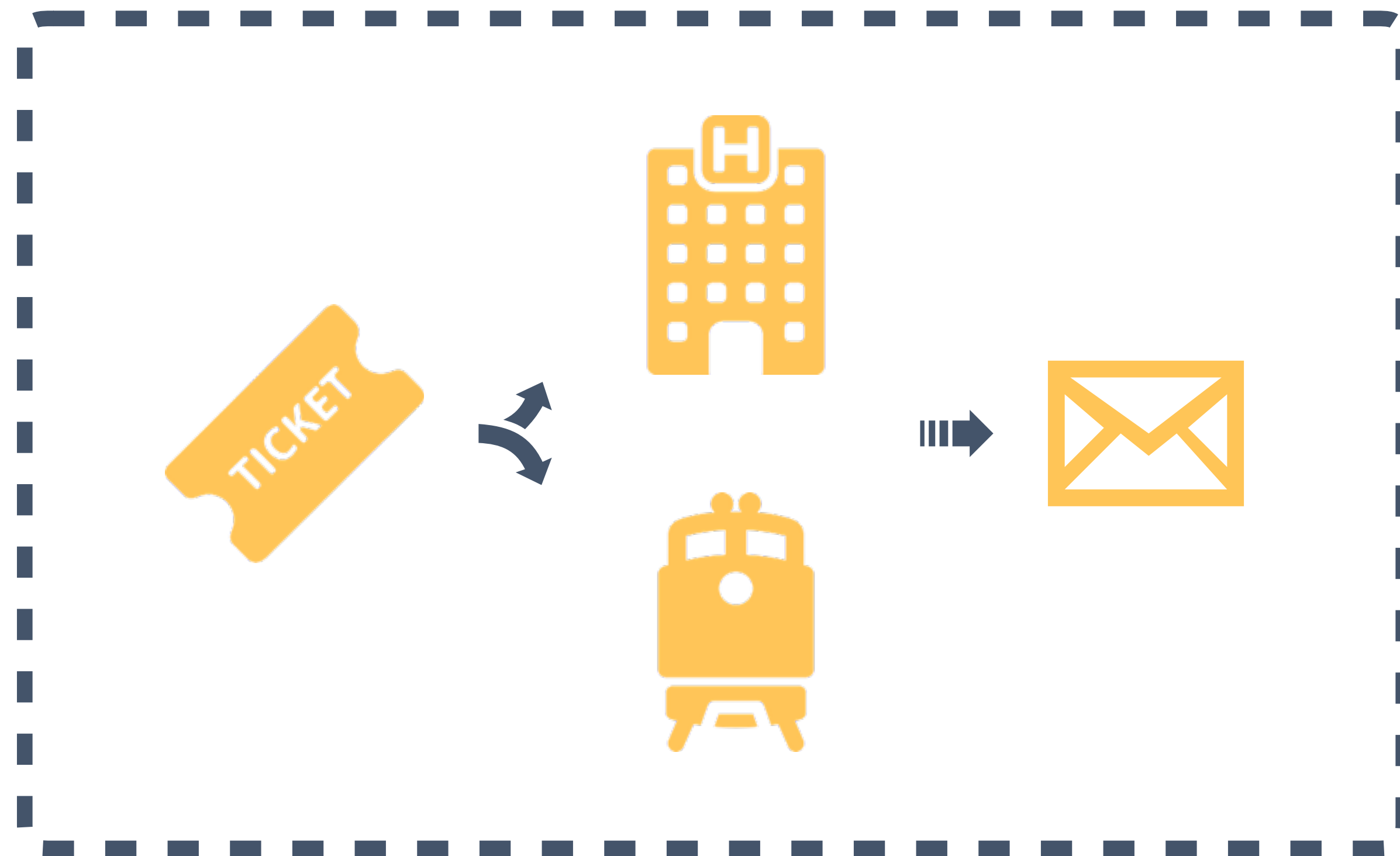
Preparing to Attend **JEEConf**

PROGRAM

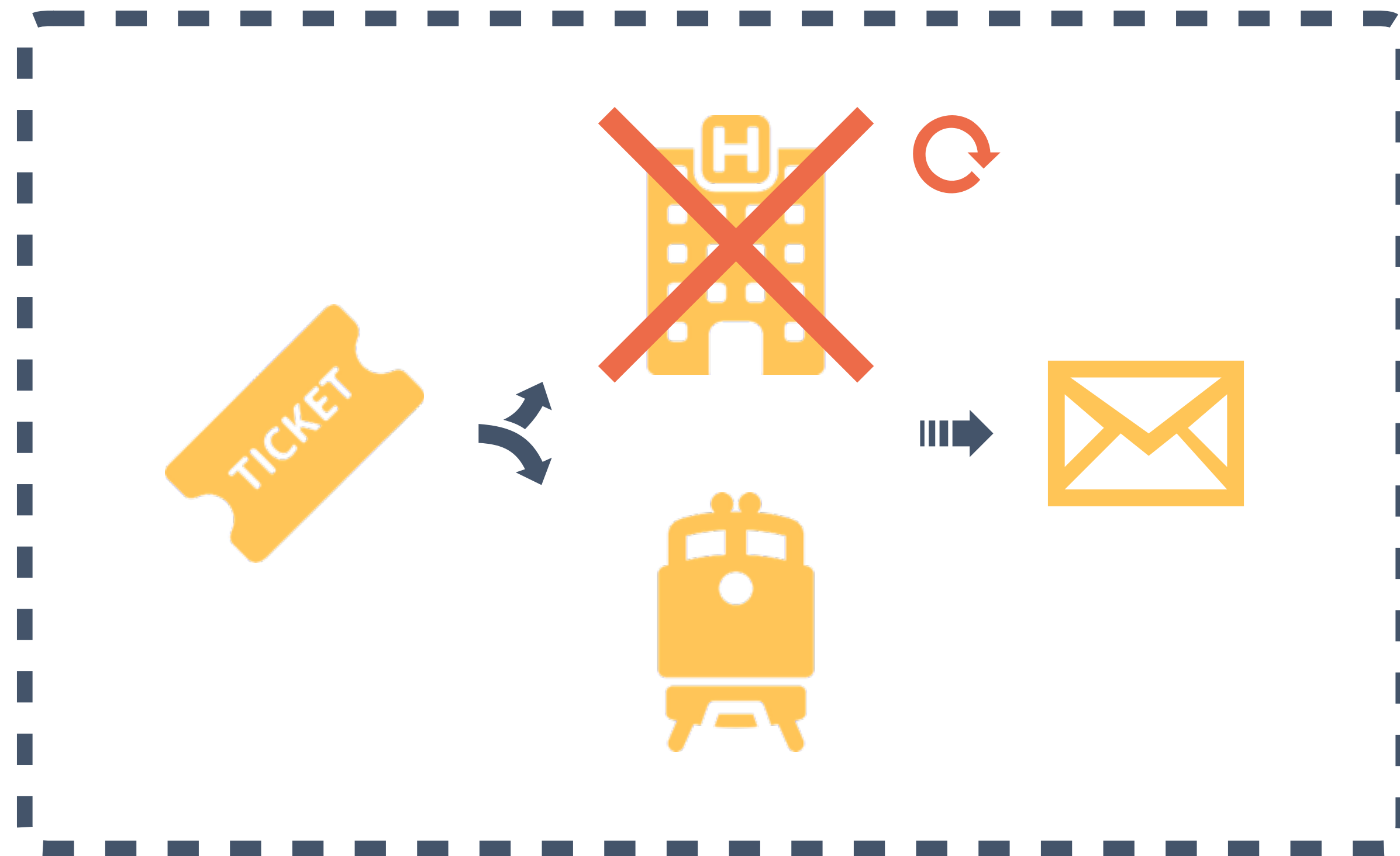


STEPS

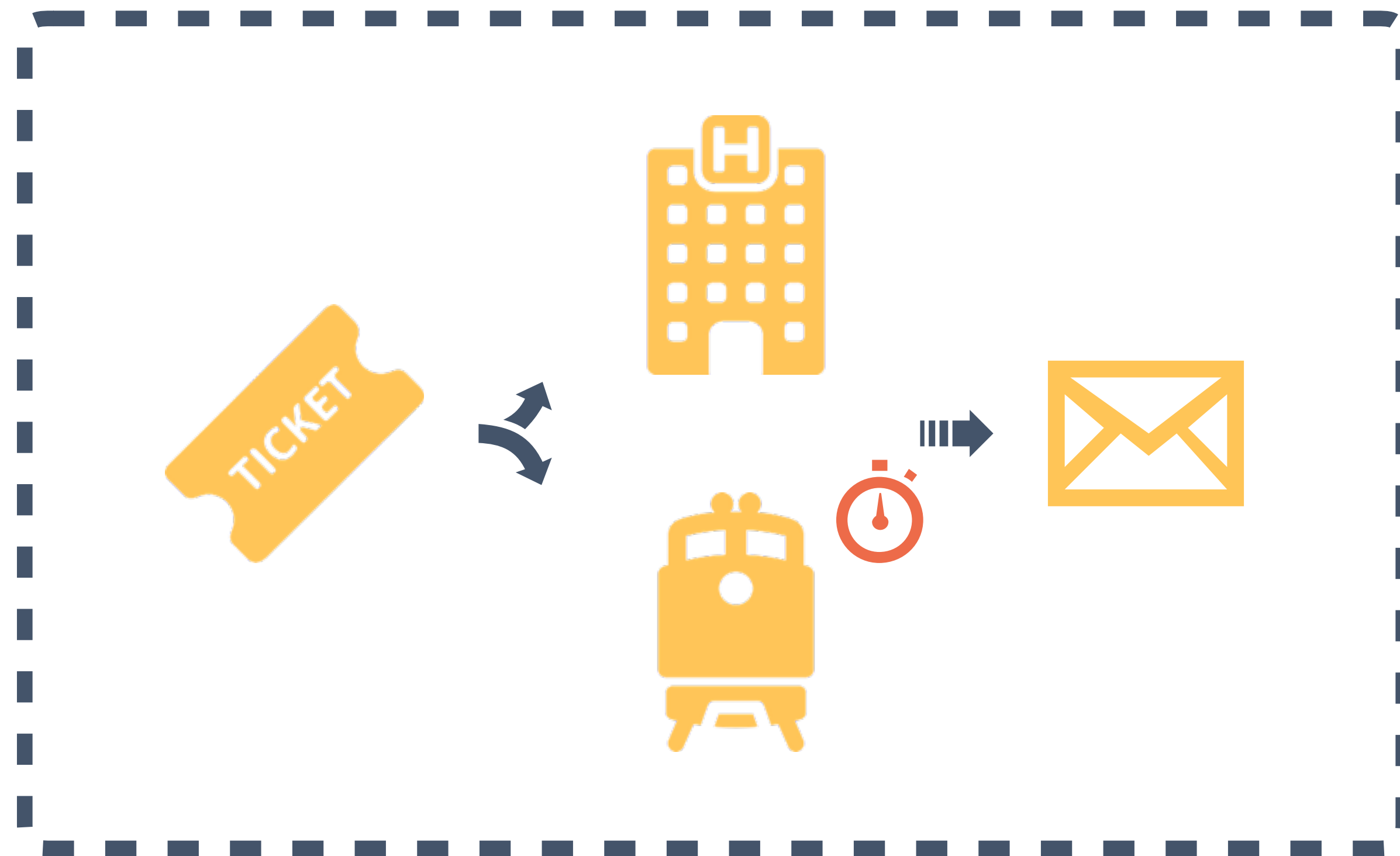
Preparing to Attend **JEEConf**



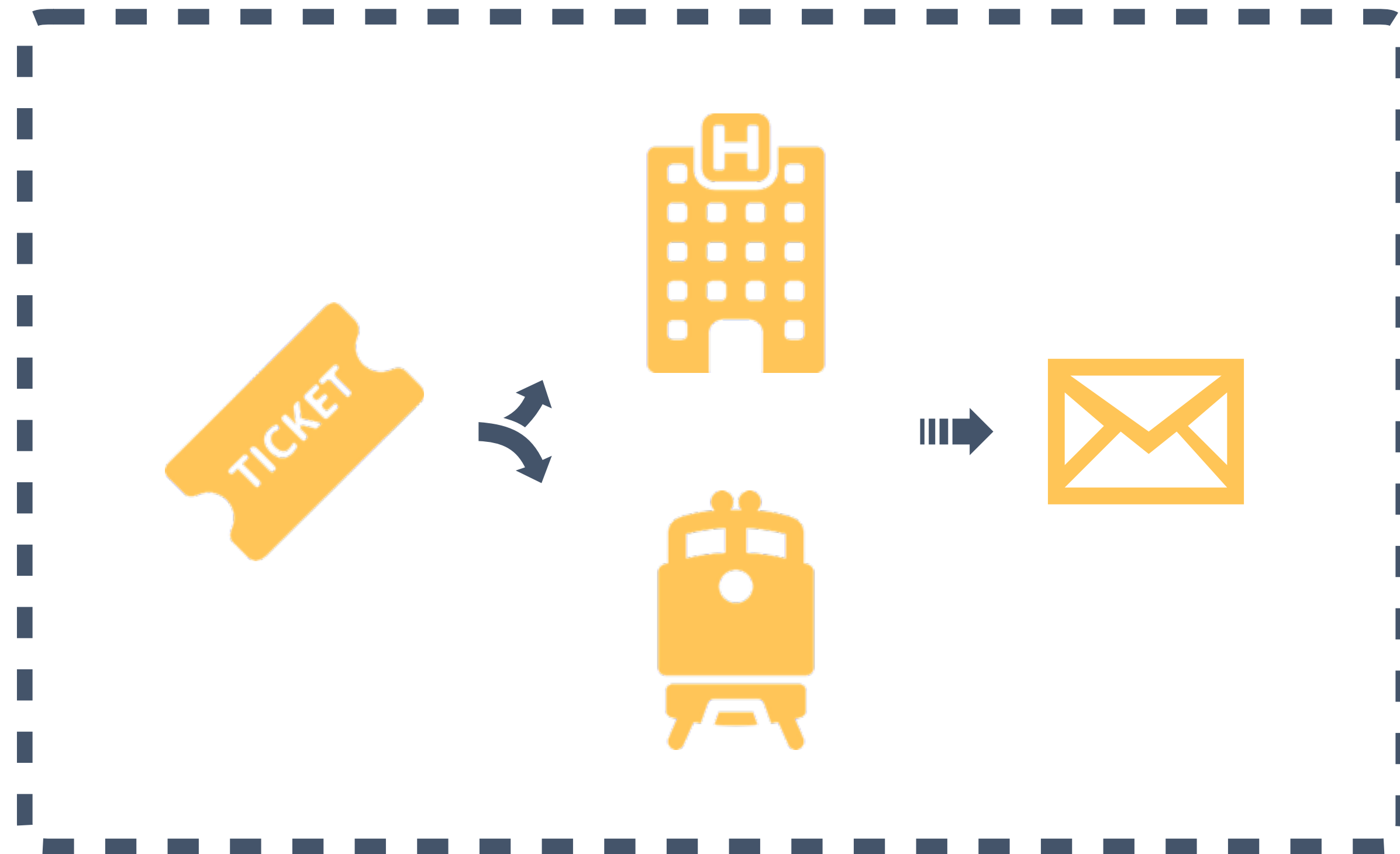
Preparing to Attend **JEEConf**



Preparing to Attend **JEEConf**



Preparing to Attend **JEEConf**

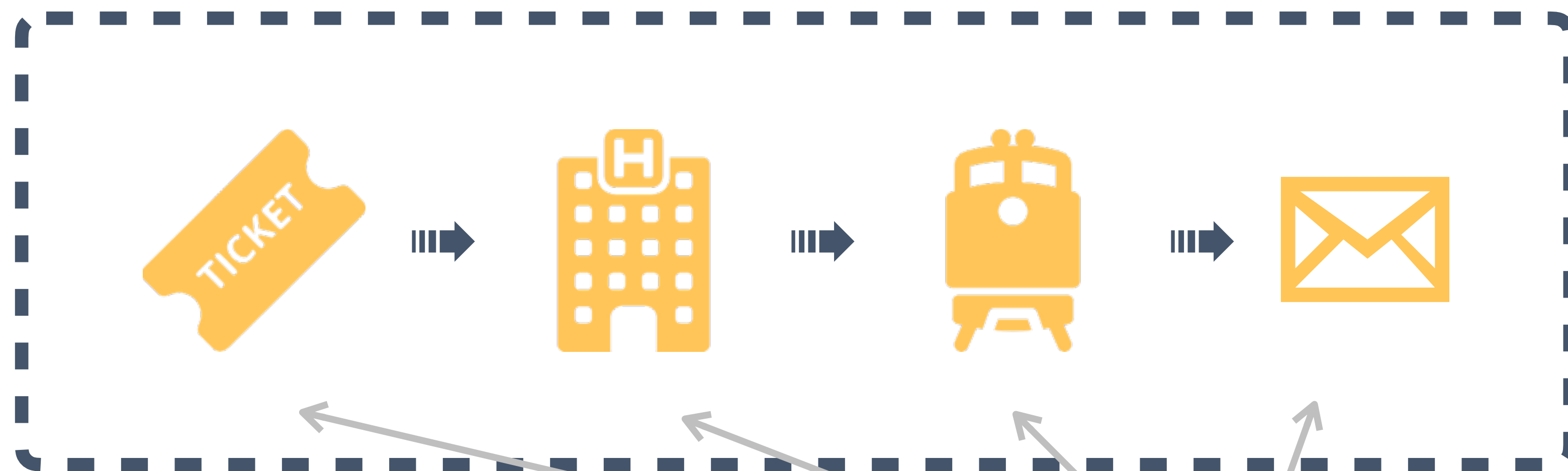




<https://aws.amazon.com/swf>

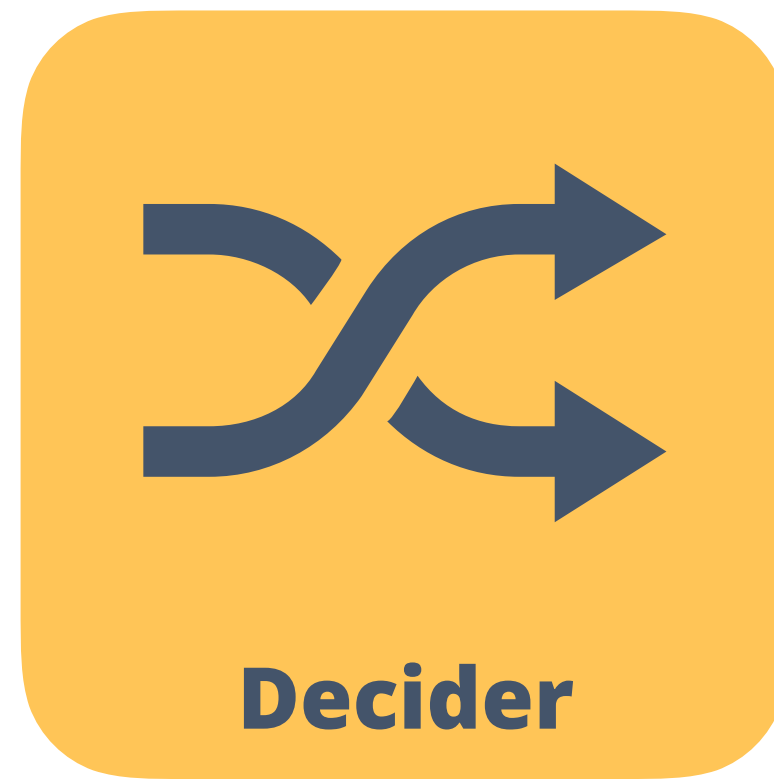
Workflows and Activities

WORKFLOW

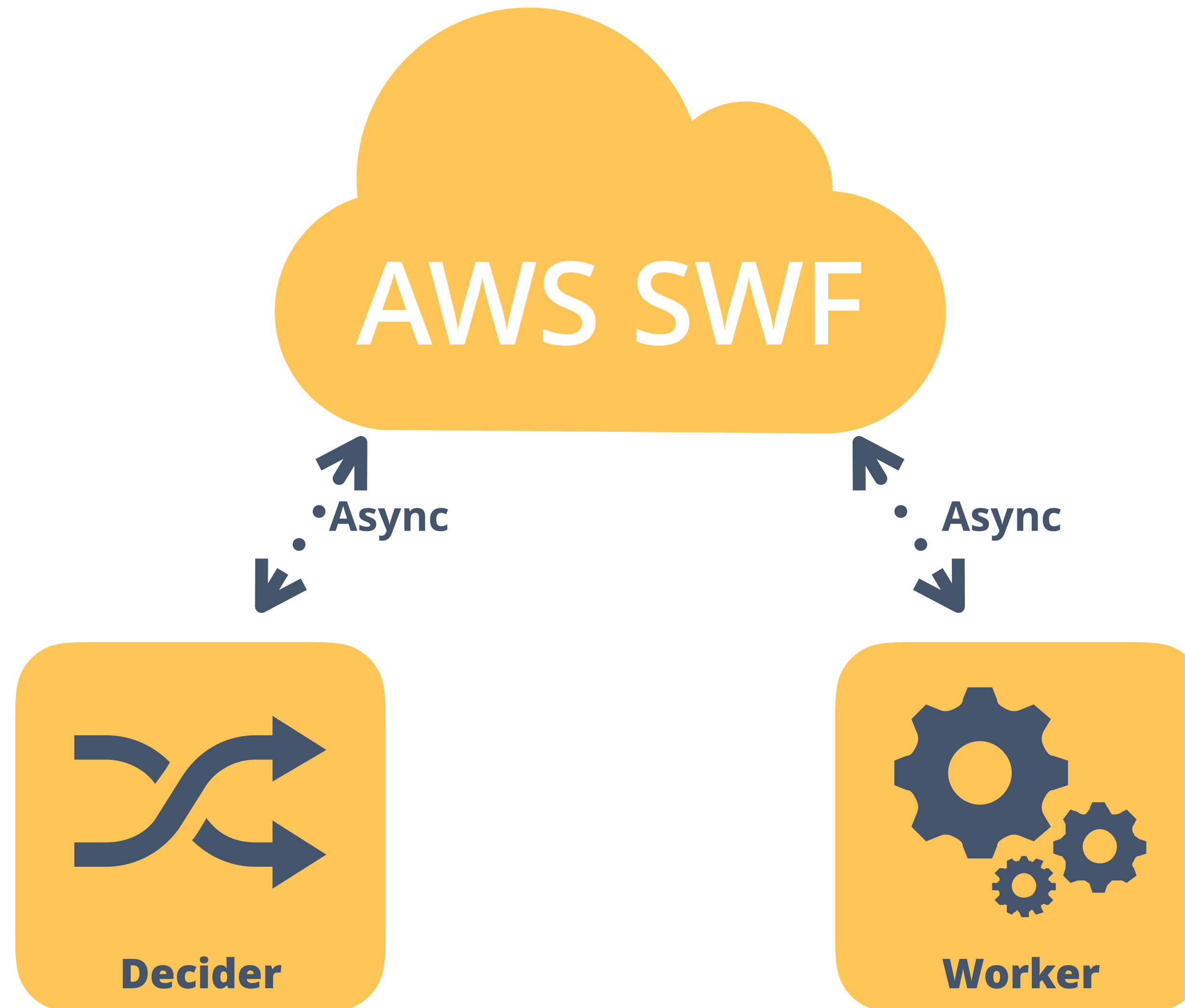


ACTIVITIES

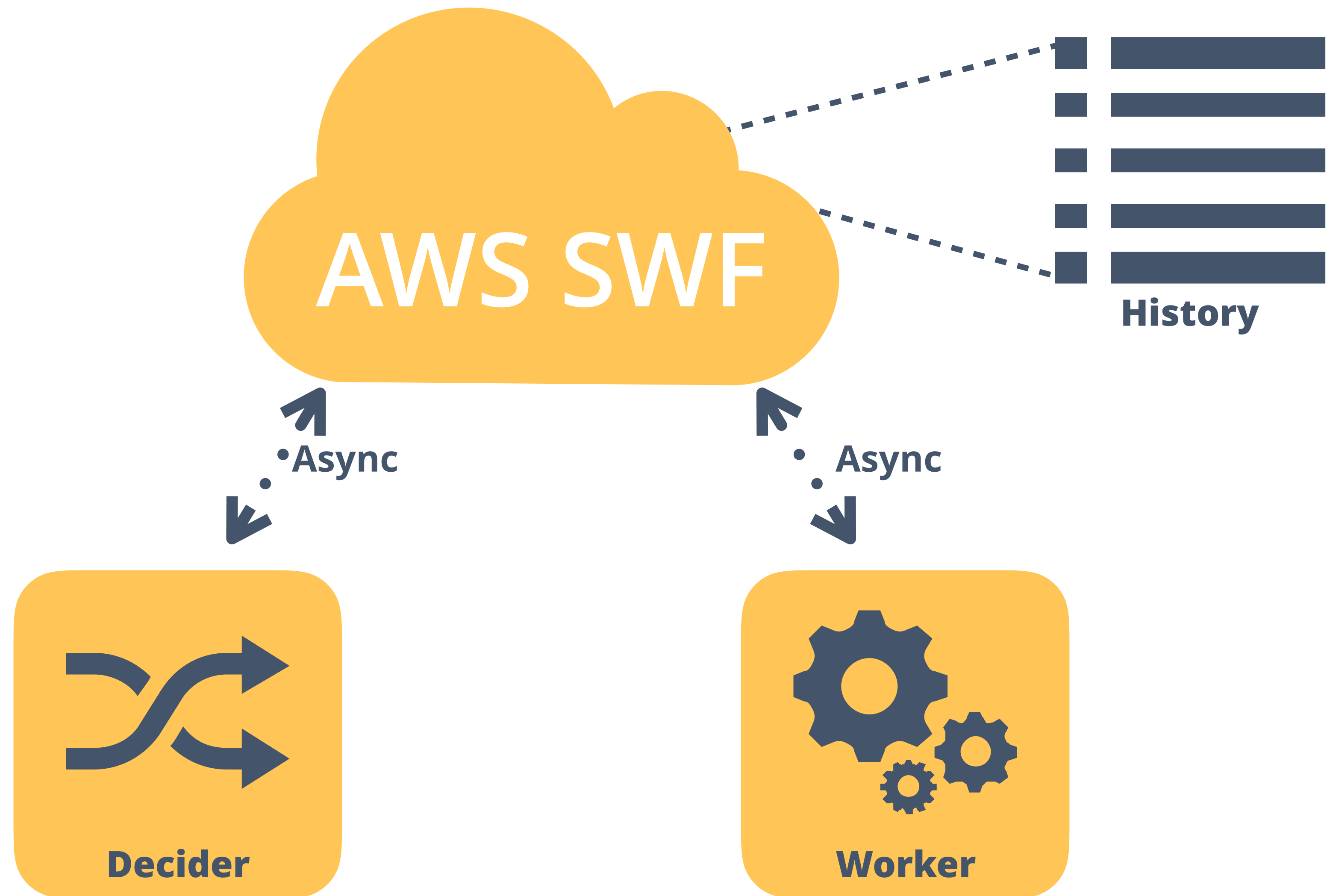
SWF Application **Architecture**



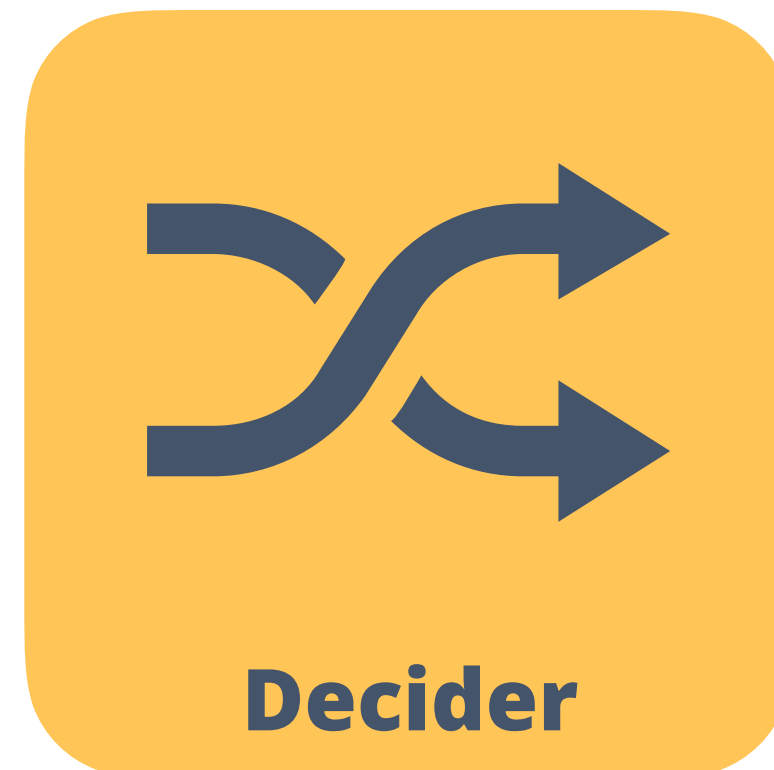
SWF Application **Architecture**



SWF Application **Architecture**



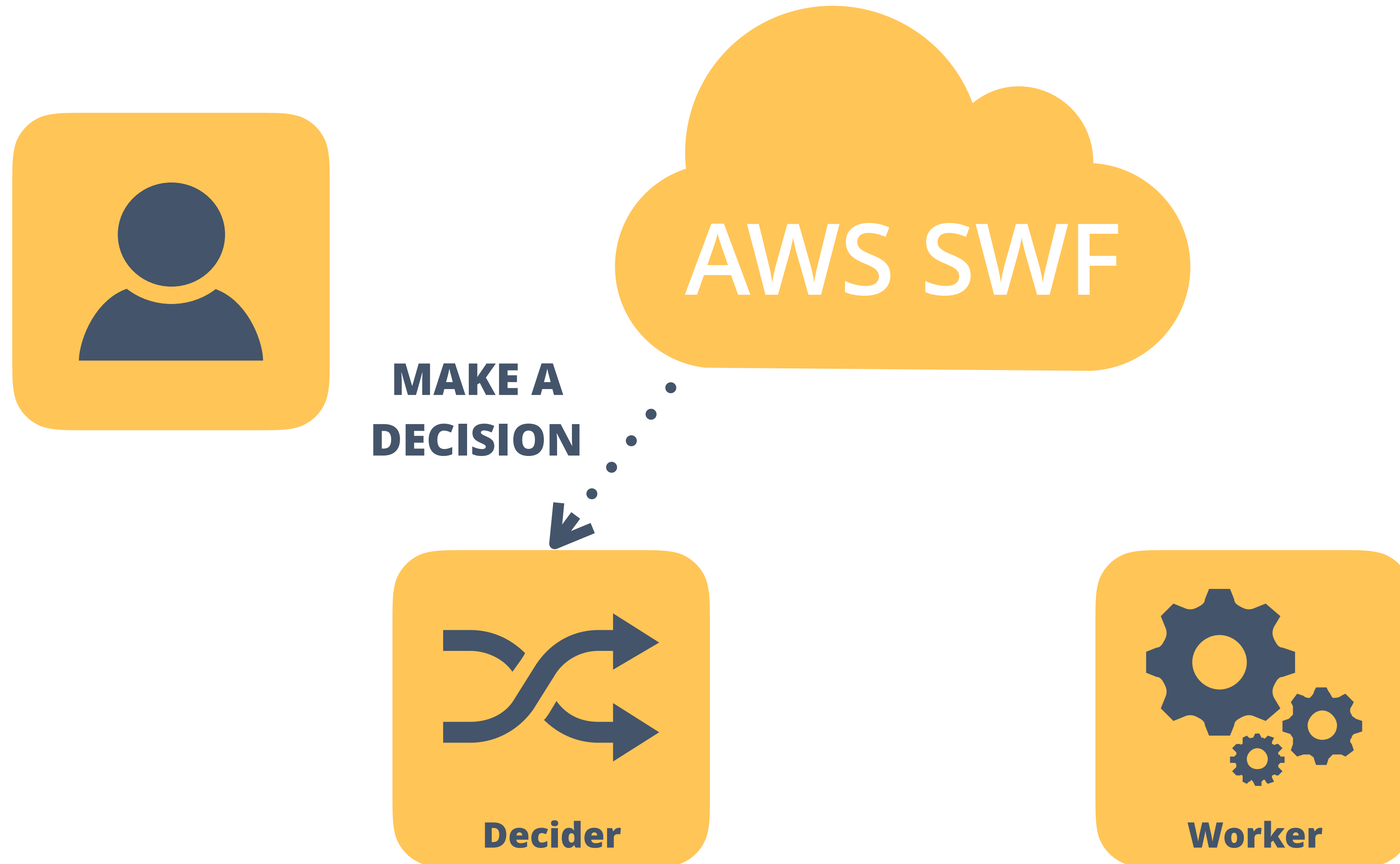
AWS SWF **Message Flow**



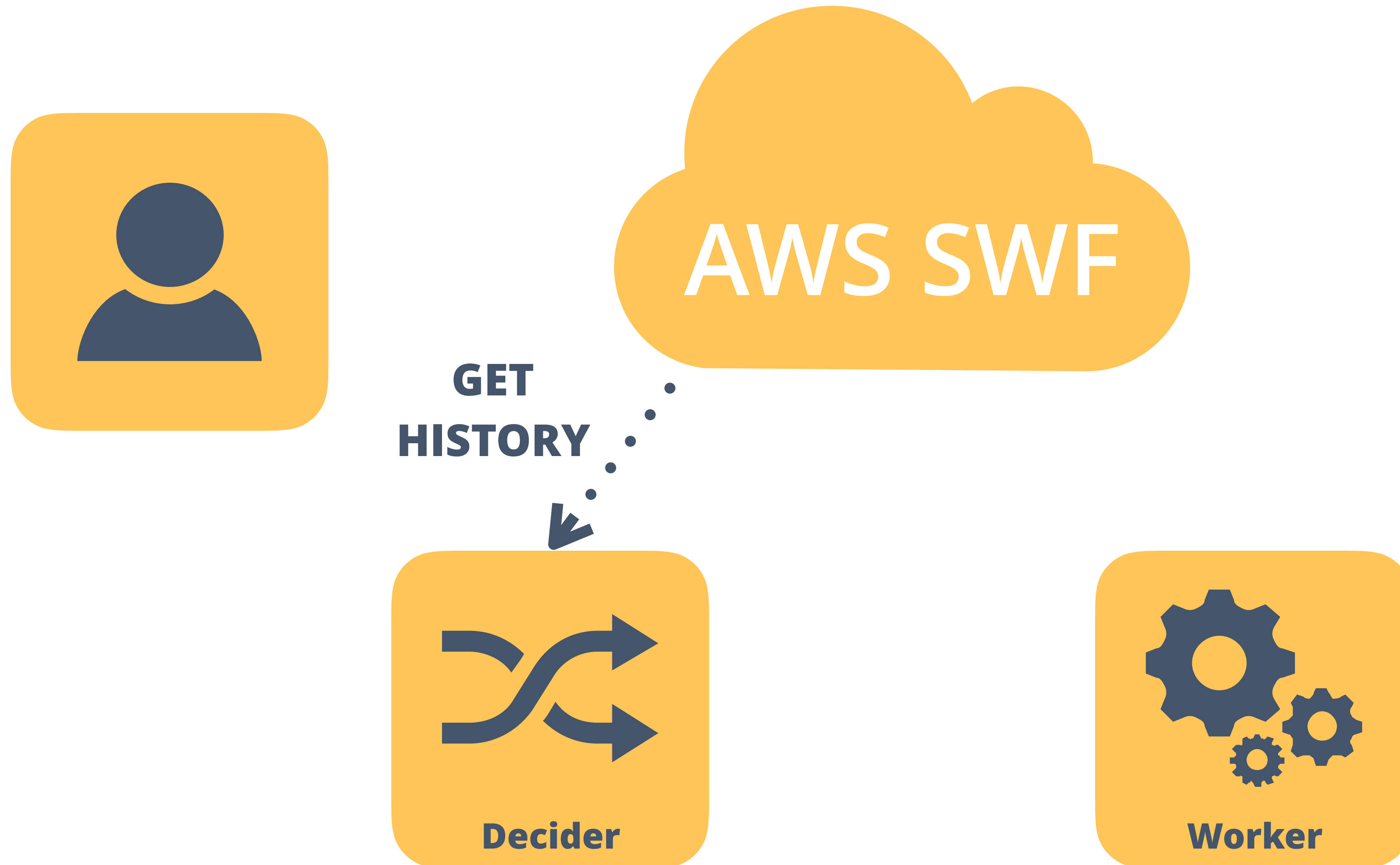
AWS SWF Message Flow



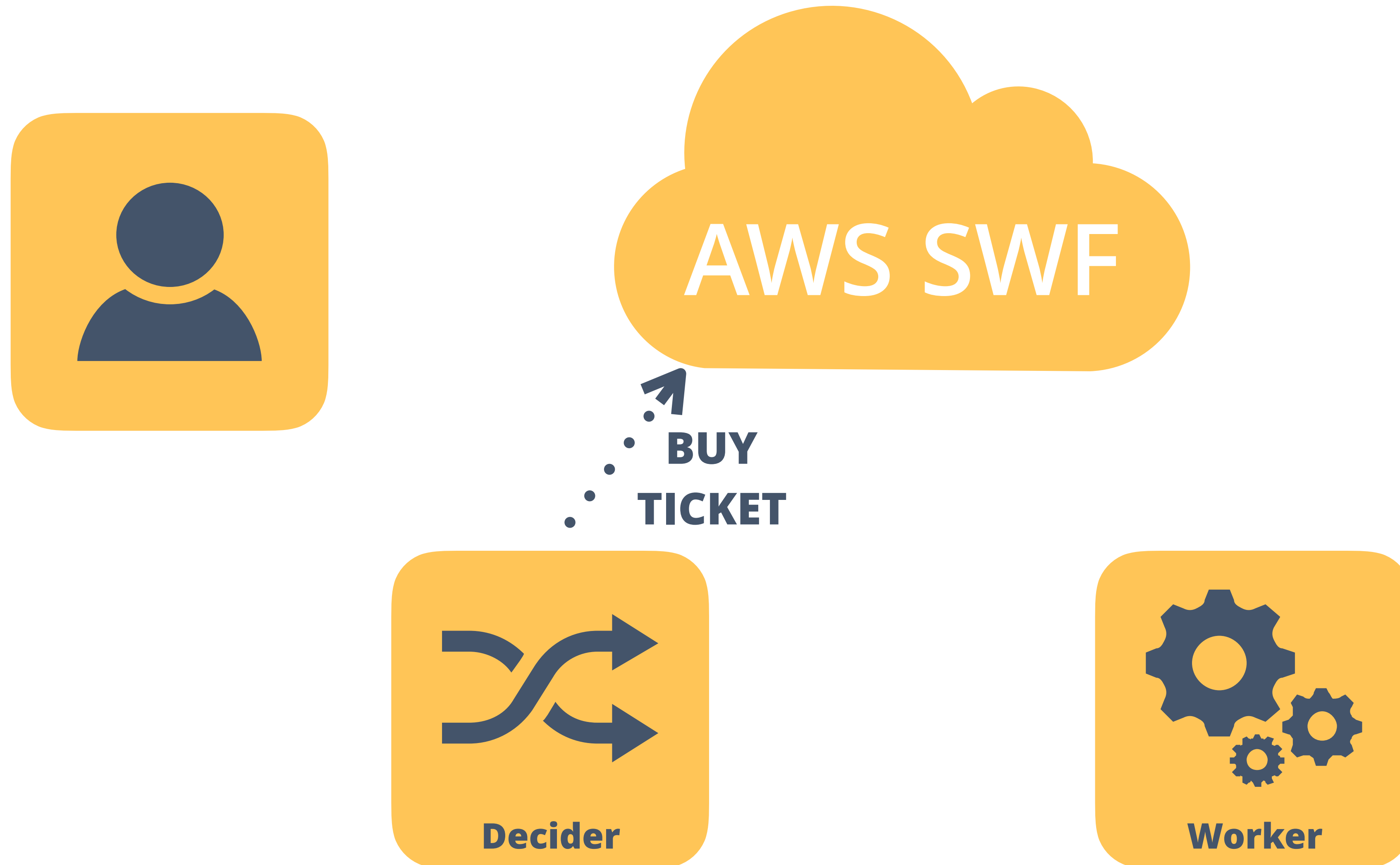
AWS SWF Message Flow



AWS SWF Message Flow



AWS SWF Message Flow



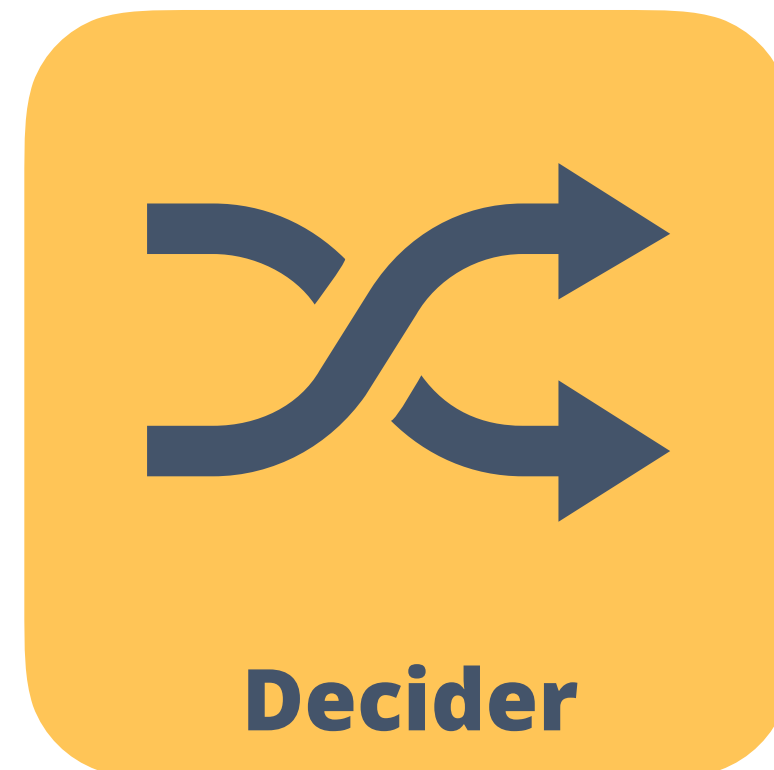
AWS SWF Message Flow



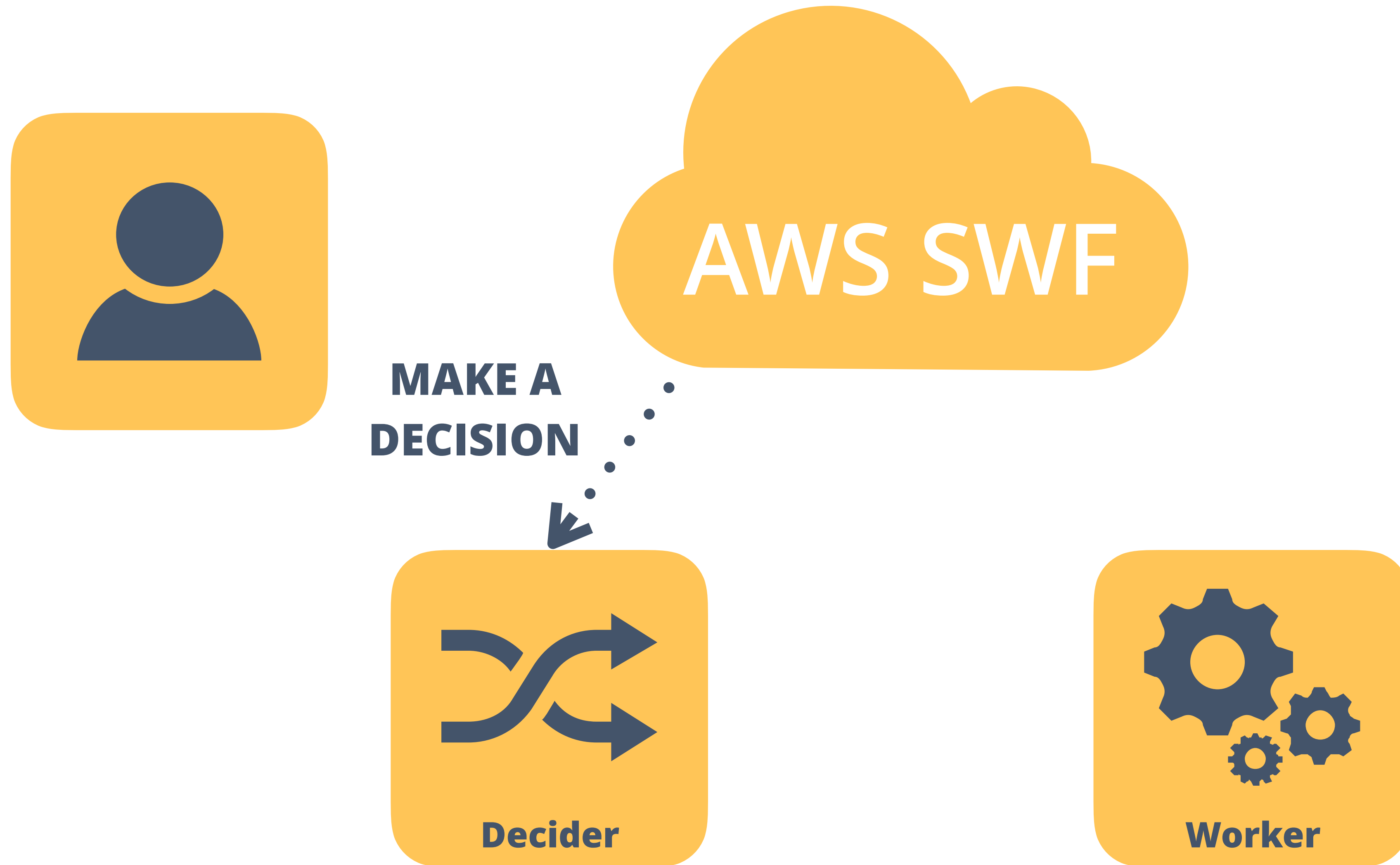
AWS SWF Message Flow



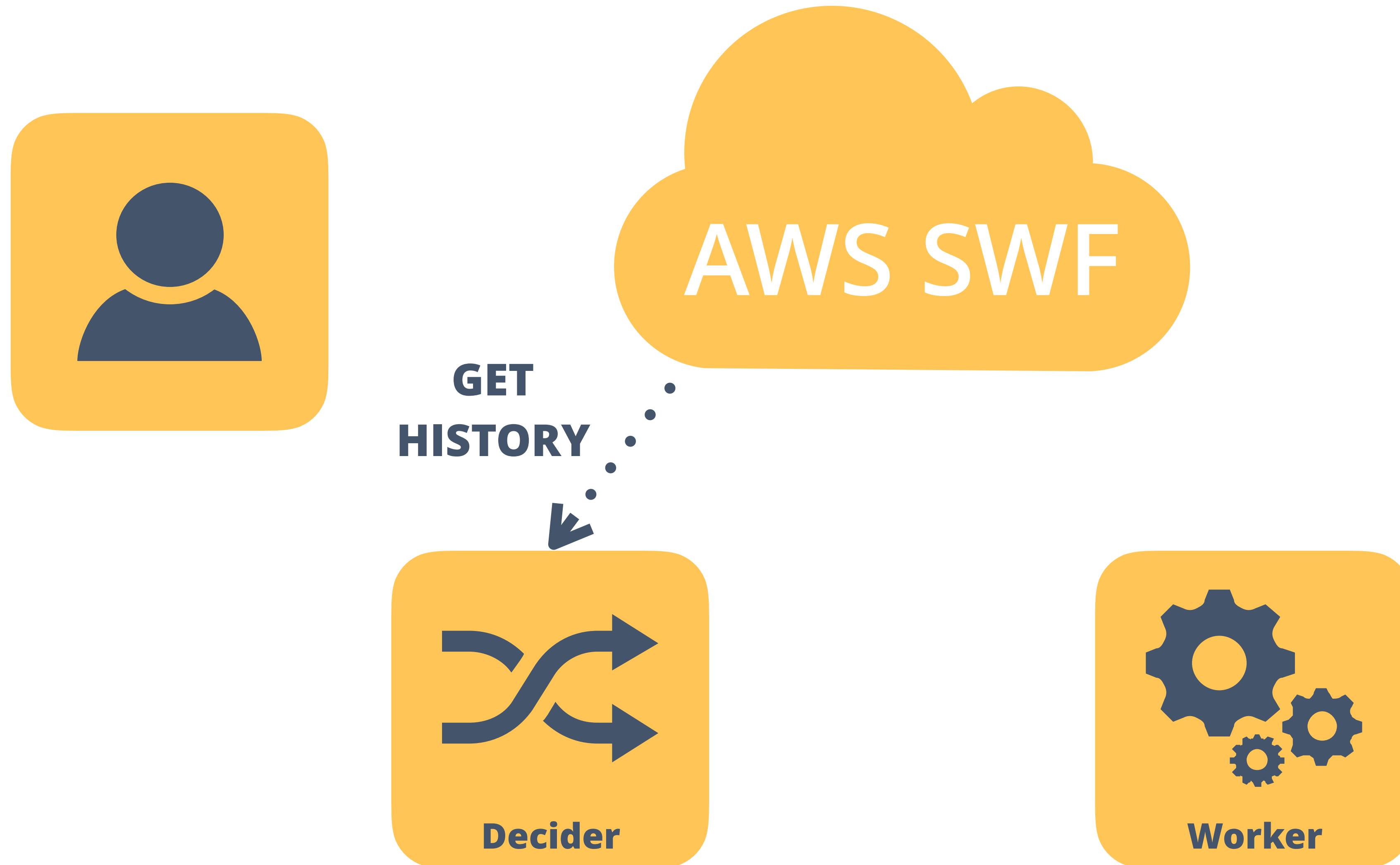
AWS SWF **Message Flow**



AWS SWF Message Flow



AWS SWF Message Flow



AWS SWF Message Flow



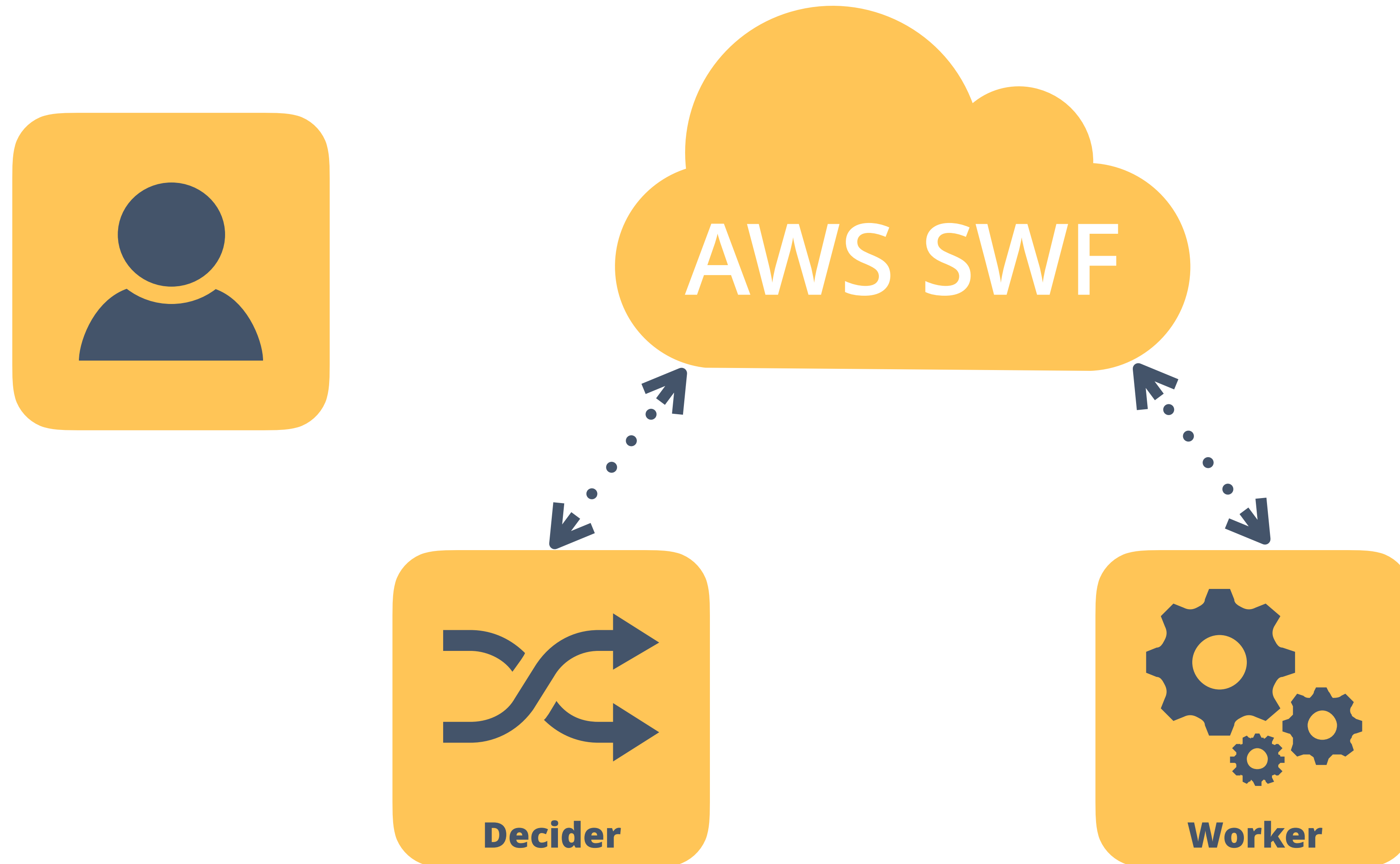
AWS SWF Message Flow



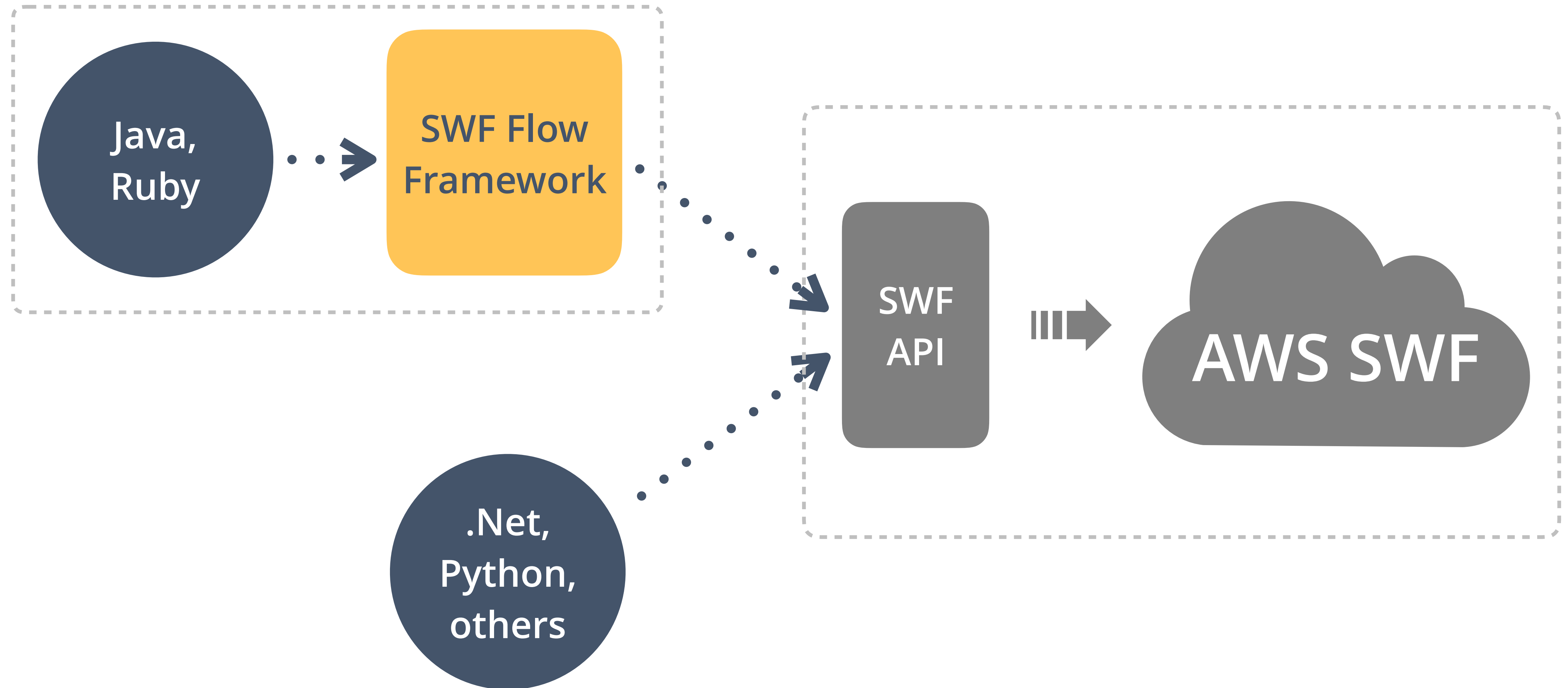
AWS SWF Message Flow



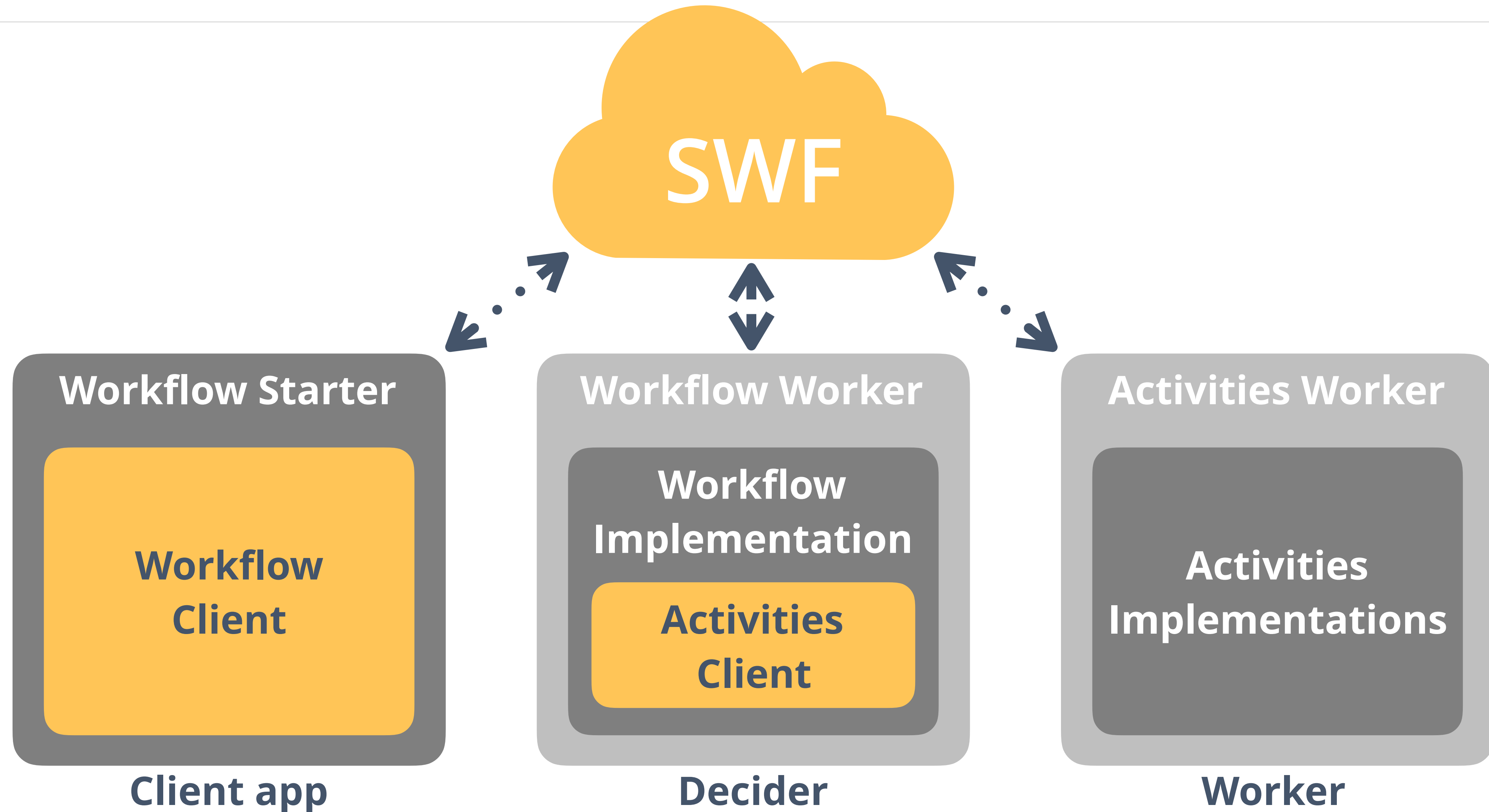
AWS SWF Message Flow



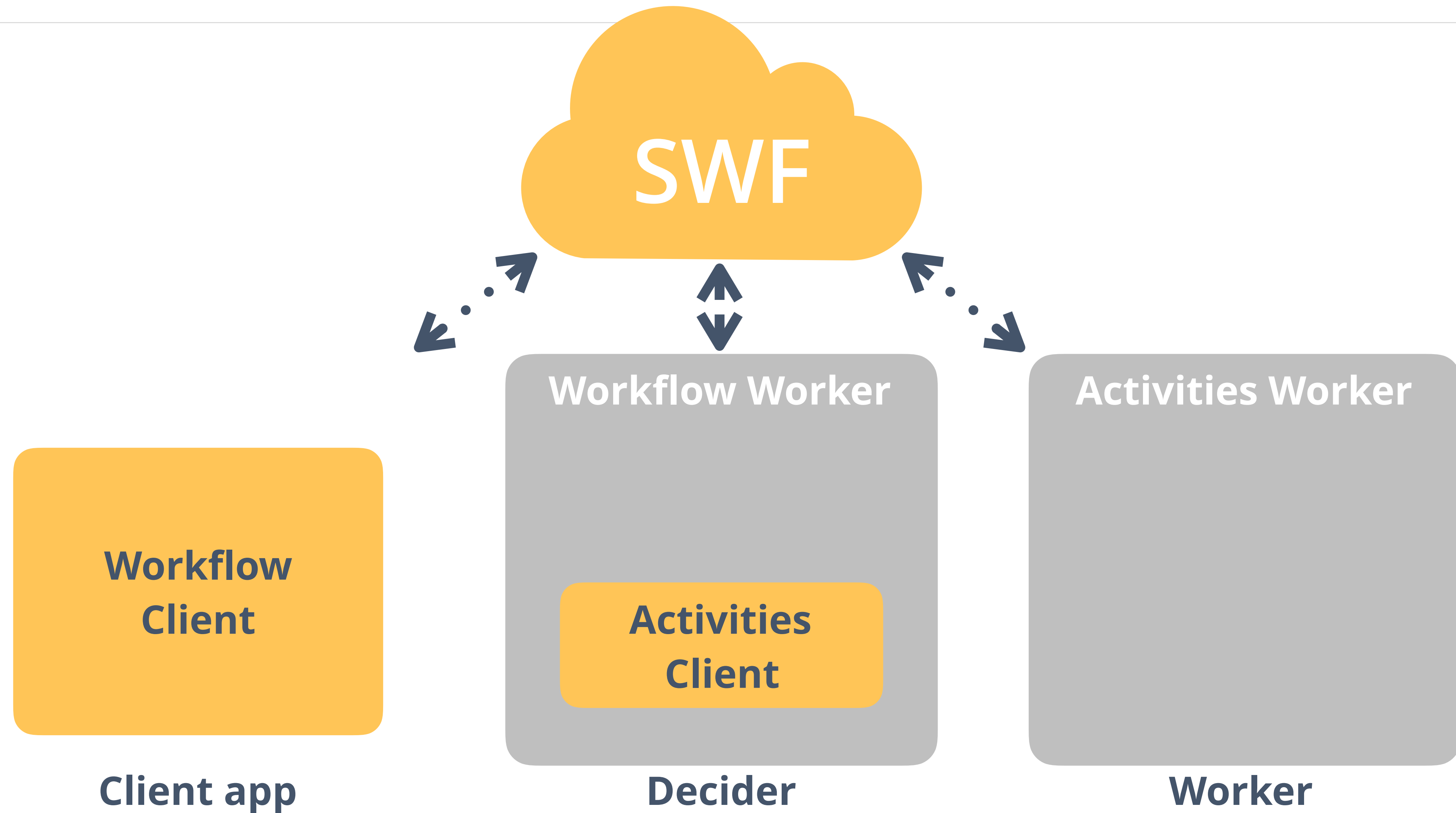
AWS SWF Flow Framework



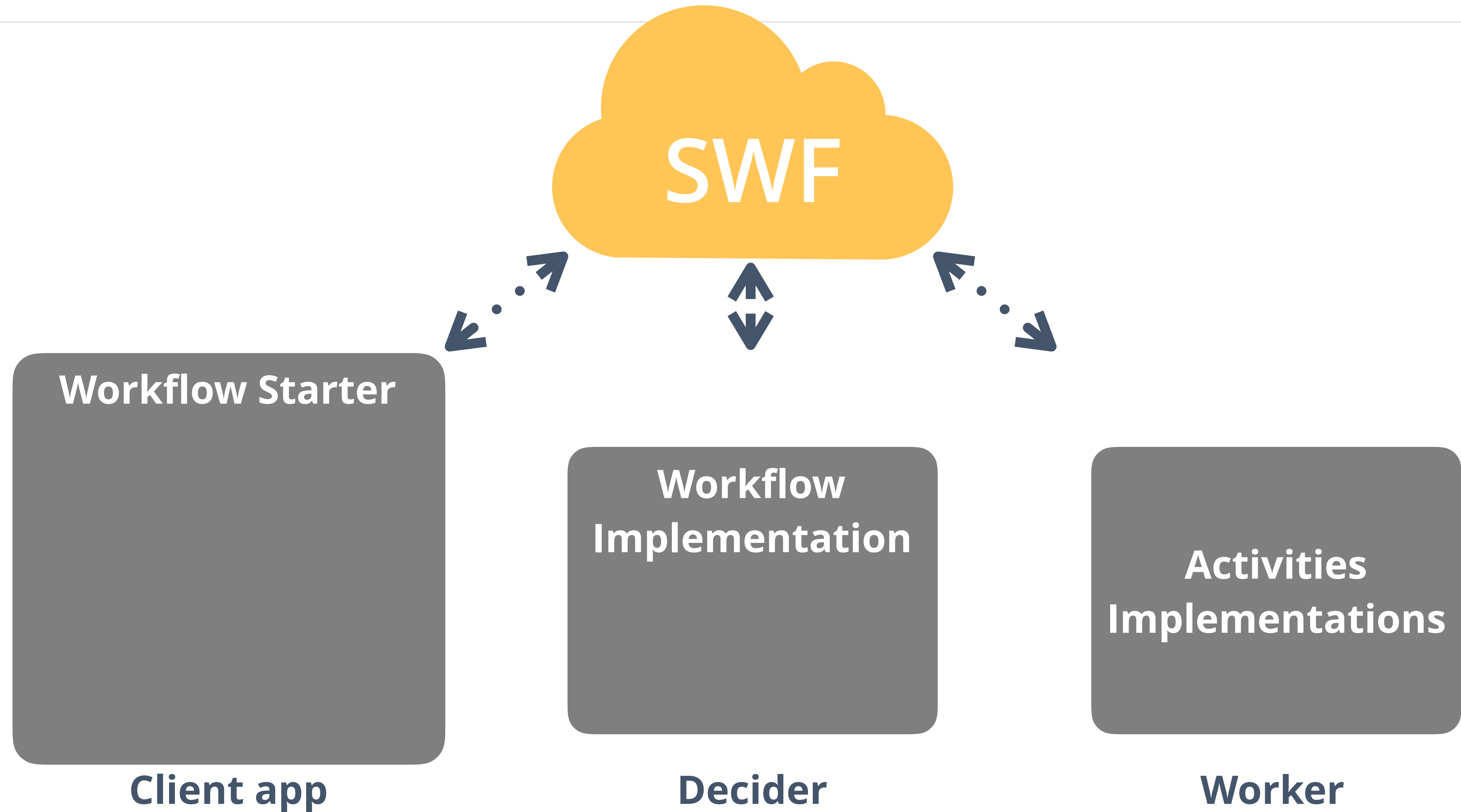
Java Flow Framework



Java Flow Framework



Java Flow Framework



Java Flow Framework **Classes**

@Workflow → Client to schedule workflows

@Activities → Client to schedule activities

Java Flow Framework **Classes**

@Workflow → Client to schedule workflows

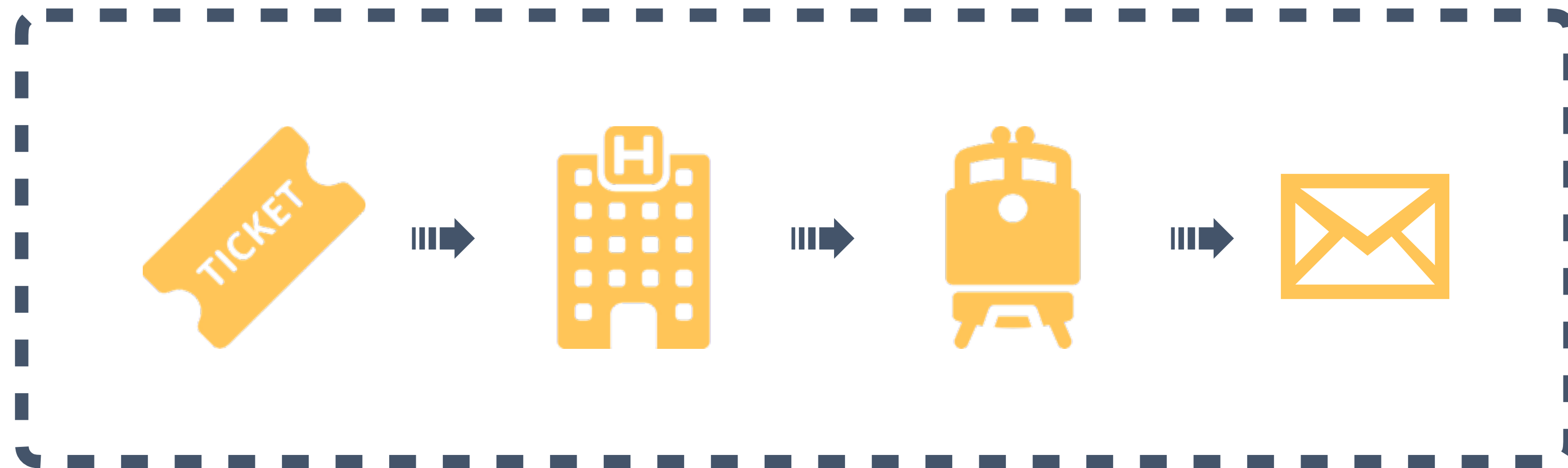
@Activities → Client to schedule activities

WorkflowWorker → Handles decision tasks

ActivitiesWorker → Handles activity tasks



Visit Organizer **Workflow**



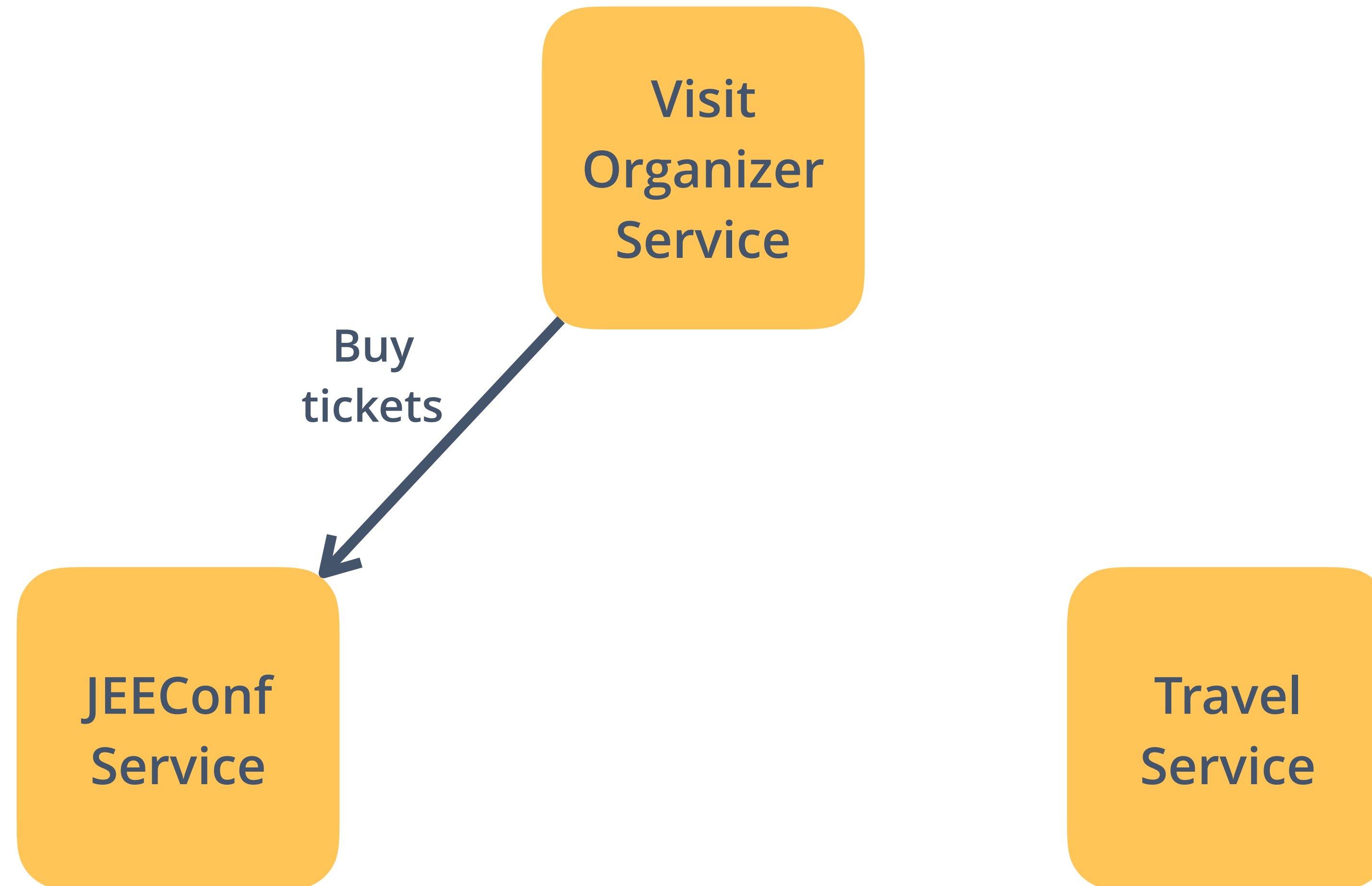
Preparing for **JEEConf**

Visit
Organizer
Service

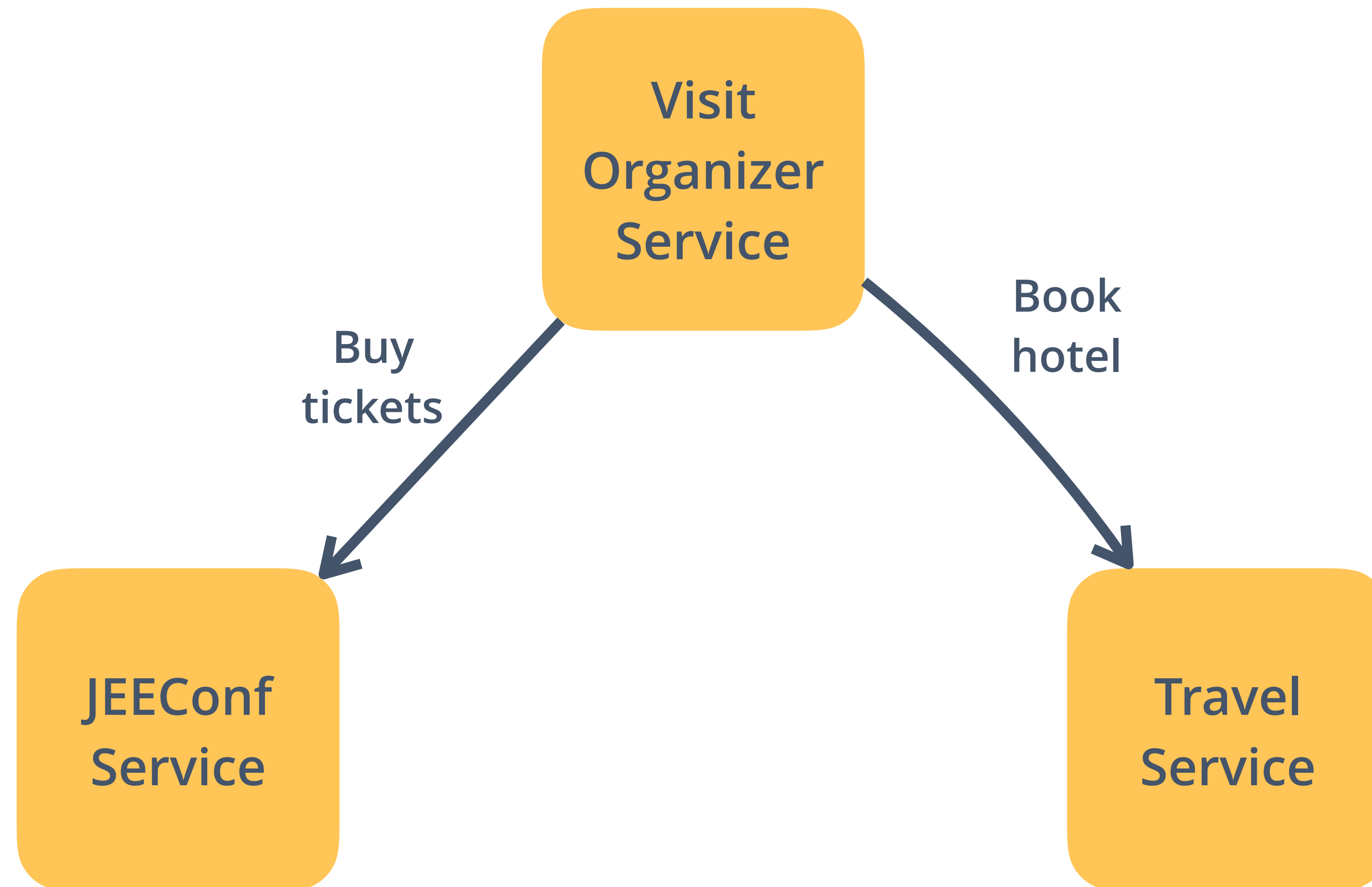
JEEConf
Service

Travel
Service

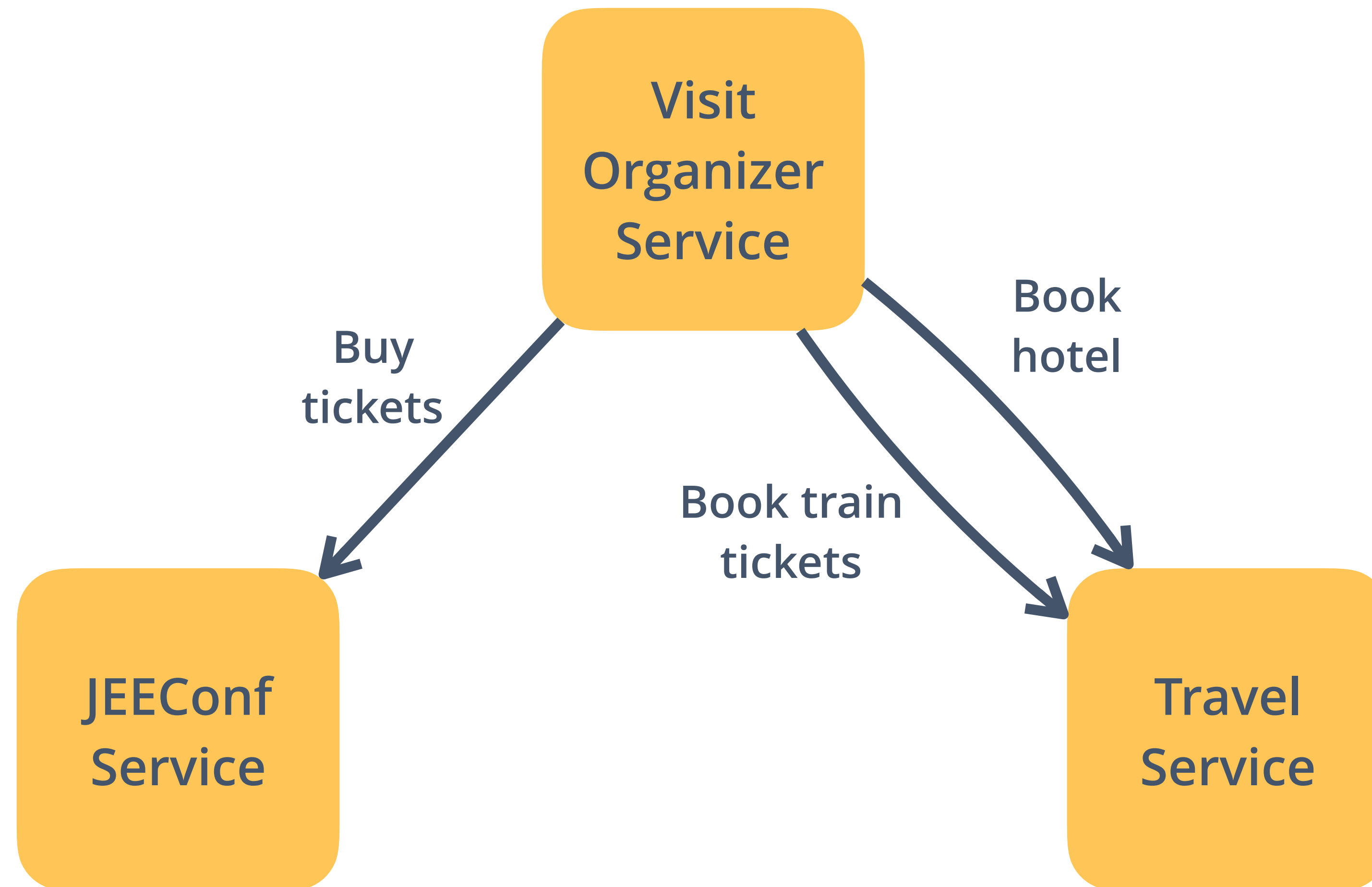
Preparing for **JEEConf**



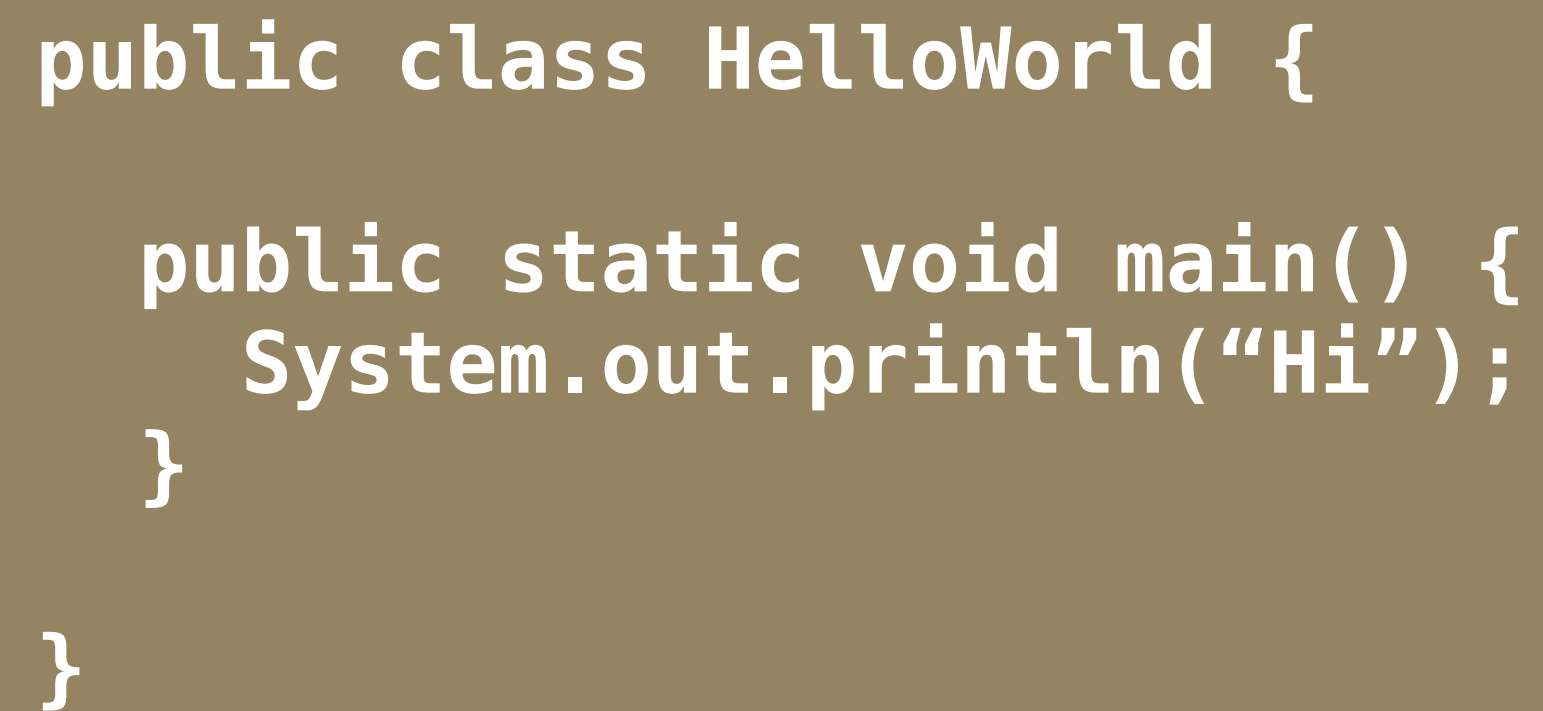
Preparing for **JEEConf**



Preparing for **JEEConf**



Time to see the code!



```
public class HelloWorld {  
  
    public static void main() {  
        System.out.println("Hi");  
    }  
  
}
```

<https://github.com/sbatyuk/aws-swf-sample>

EASY, RIGHT?



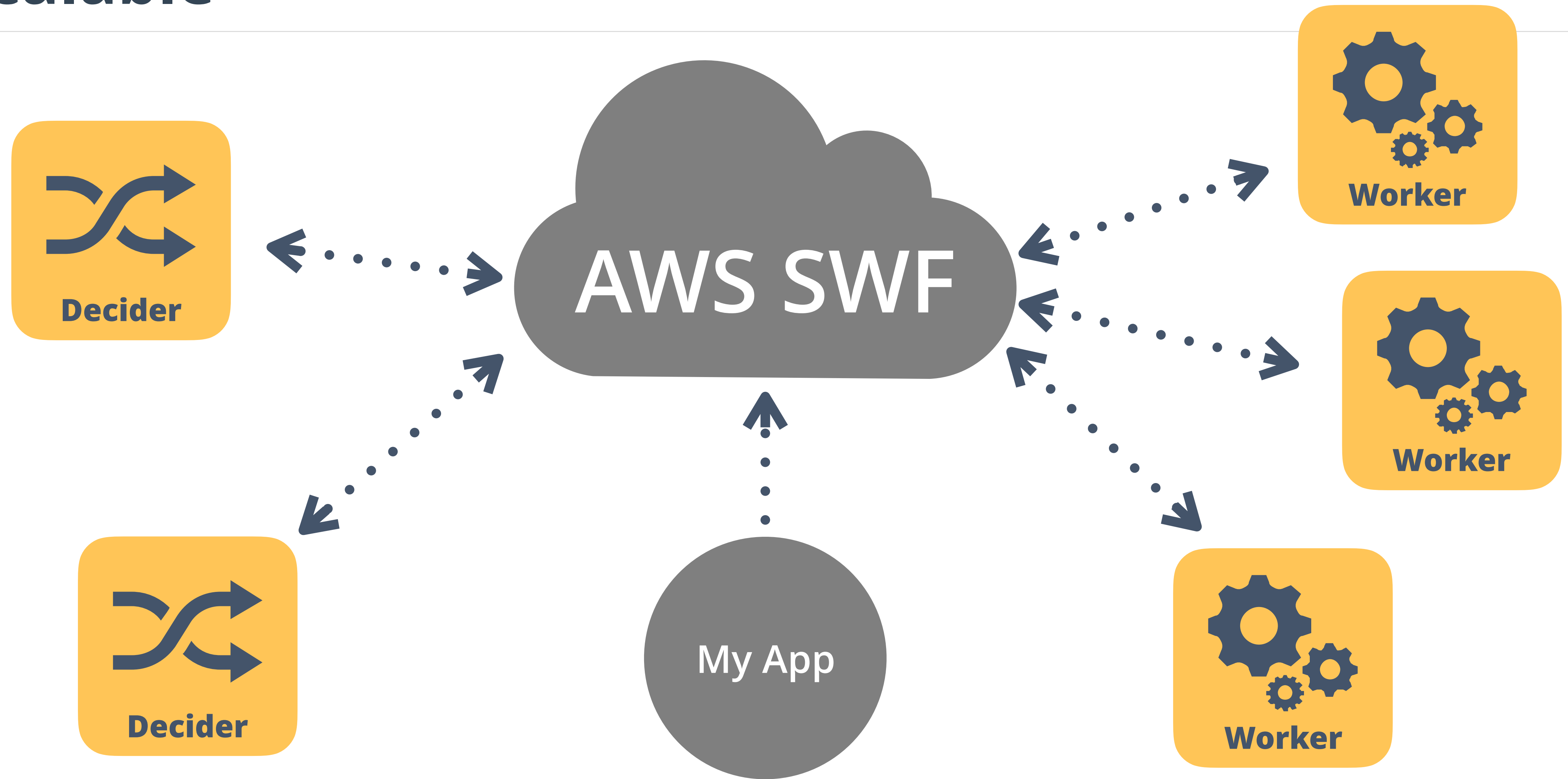


Features

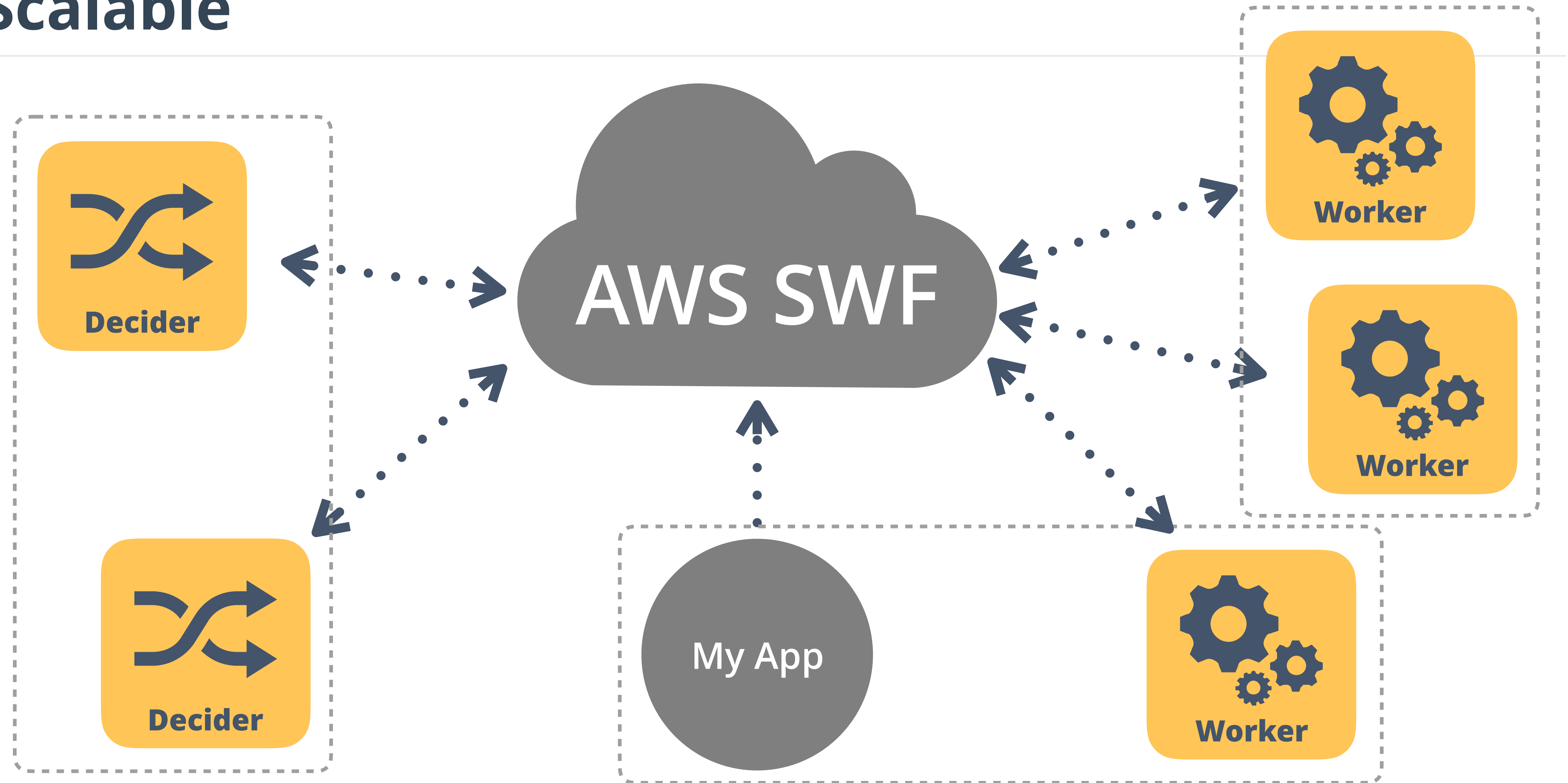
Fully Managed Service



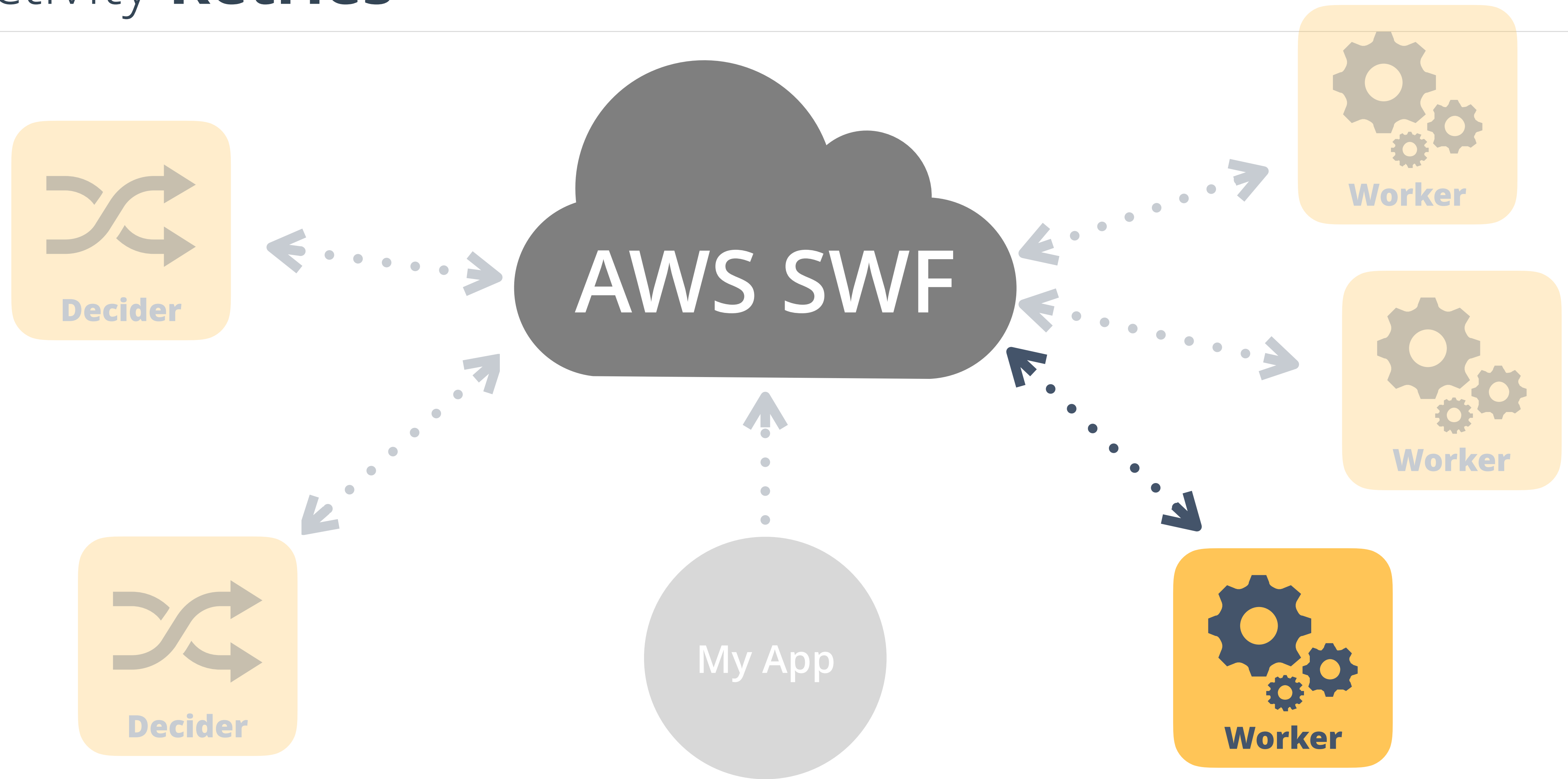
Scalable



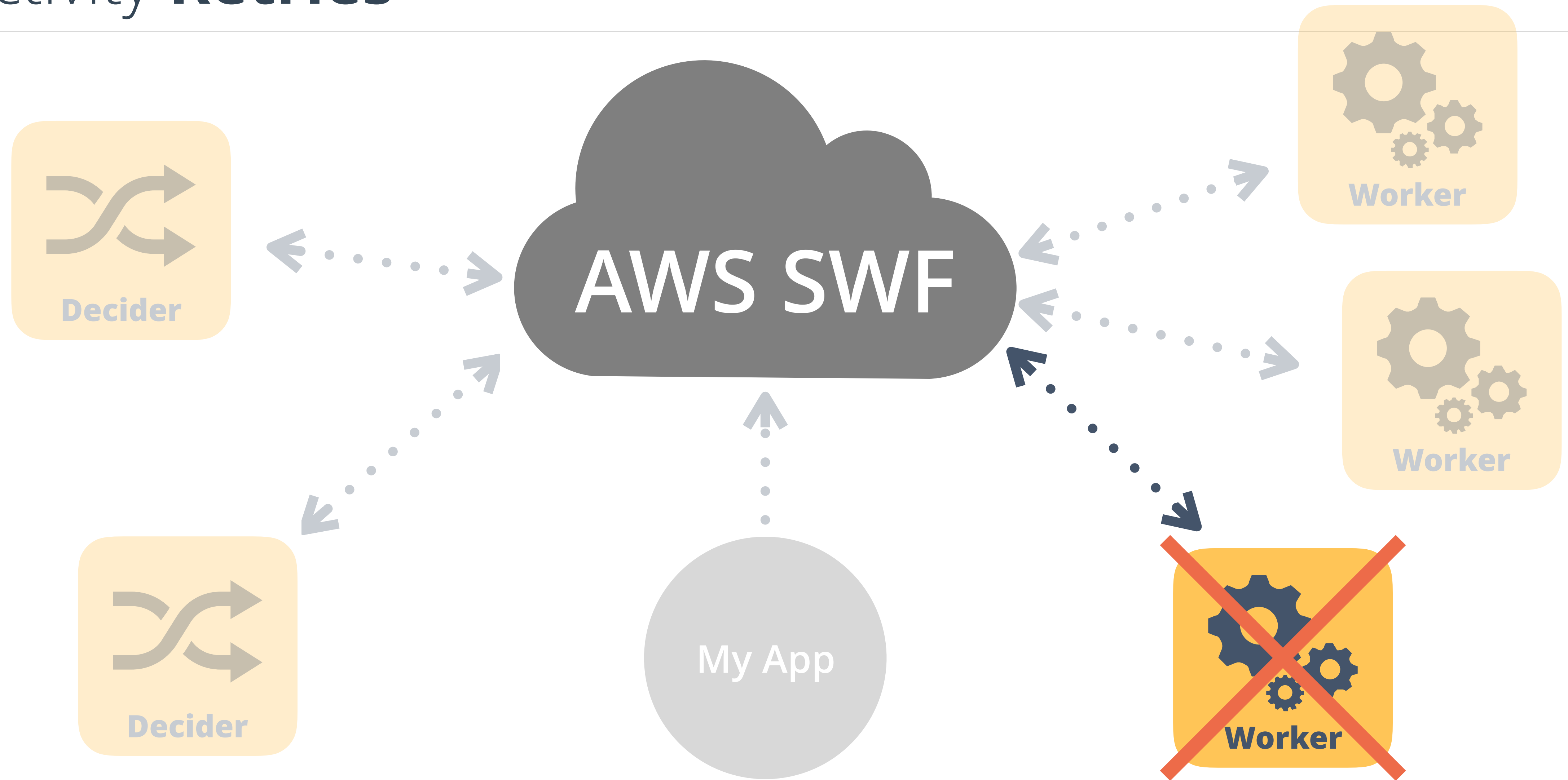
Scalable



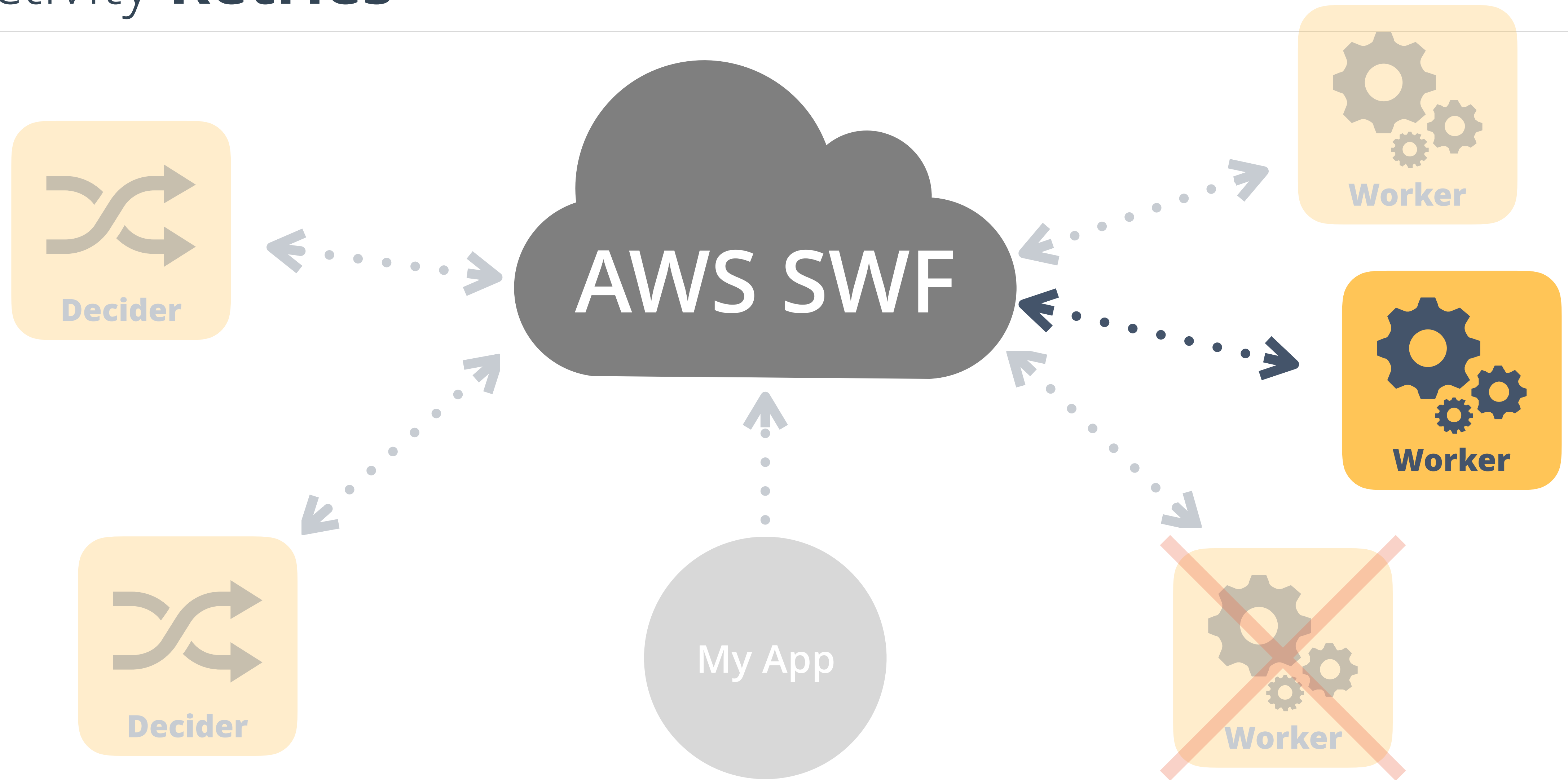
Activity **Retries**



Activity **Retries**



Activity **Retries**



Workflow Execution History

SWF Management Console

Serhiy

← → ↺

https://eu-central-1.console.aws.amazon.com/swf/home?region=eu-central-1#execution_activities:domain=JEEConf;workflowId=f7...

☆ ⓘ ⌵ ☰

📦

AWS

▼

Services

▼

📦

SWF

🔵

DynamoDB

📡

Kinesis

📦

CloudForr

Edit

▼

seldon @ seldon

▼

Frankfurt

▼

Support

▼

Workflow Execution: f74aefa8-90d4-4492-95f1-e582b5429f99

Domain: JEEConf

↺ ⓧ

▶

Summary

Events

Activities

Activities 1 to 3

Activity ID	Name	Version	Status	Schedule Time	Start Time	End Time
2	TravelService.bookHotel	1.0	Completed	Wednesday, May 11, 2016 9:59:31 AM UTC+3	Wednesday, May 11, 2016 9:59:31 AM UTC+3	Wednesday, May 11, 2016 9:59:31 AM UTC+3
3	TravelService.bookTrainTickets	1.0	Completed	Wednesday, May 11, 2016 9:59:31 AM UTC+3	Wednesday, May 11, 2016 9:59:31 AM UTC+3	Wednesday, May 11, 2016 9:59:31 AM UTC+3
1	JEEConfService.buyTicket	1.0	Completed	Wednesday, May 11, 2016 9:59:31 AM UTC+3	Wednesday, May 11, 2016 9:59:31 AM UTC+3	Wednesday, May 11, 2016 9:59:31 AM UTC+3

TravelService.bookHotel [with Activity ID 2] selected

Control:

Not Specified

Decision Task Completed Event Id:

13

Heartbeat Timeout:

NONE

Input:

["Ljava.lang.Object;","Serhiy"]

Schedule To Close Timeout:

NONE

Schedule To Start Timeout:

10 seconds

💬

Feedback

🌐

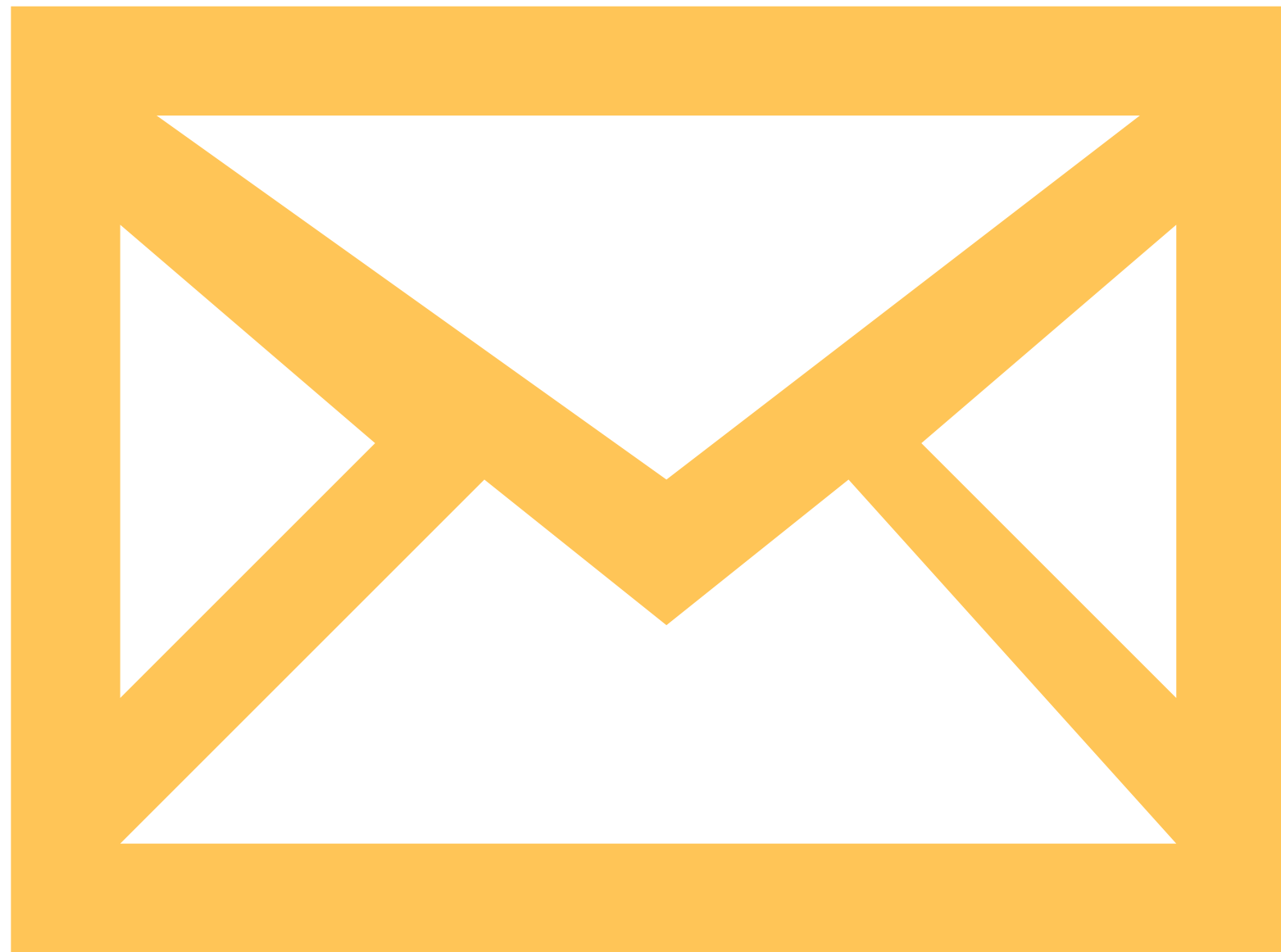
English

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Exactly Once Delivery



= 1

Integration with **Spring** and **JUnit**



JUnit

AWS SWF Pricing



10,000 workflows in a
day with 3 activities each:

\$1.75



Lessons Learnt

Workflow Method **Replays**

@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #1**

@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #1**

Get History

@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #1**

Schedule



@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #1**

Skip

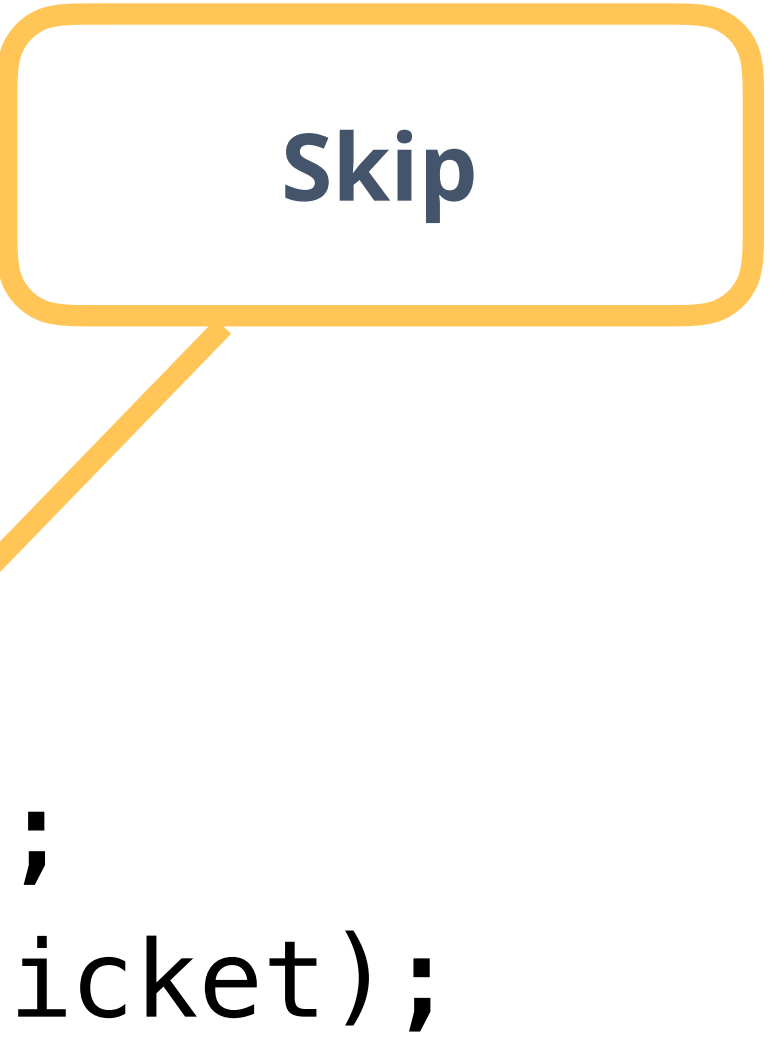
@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #1**

@Override

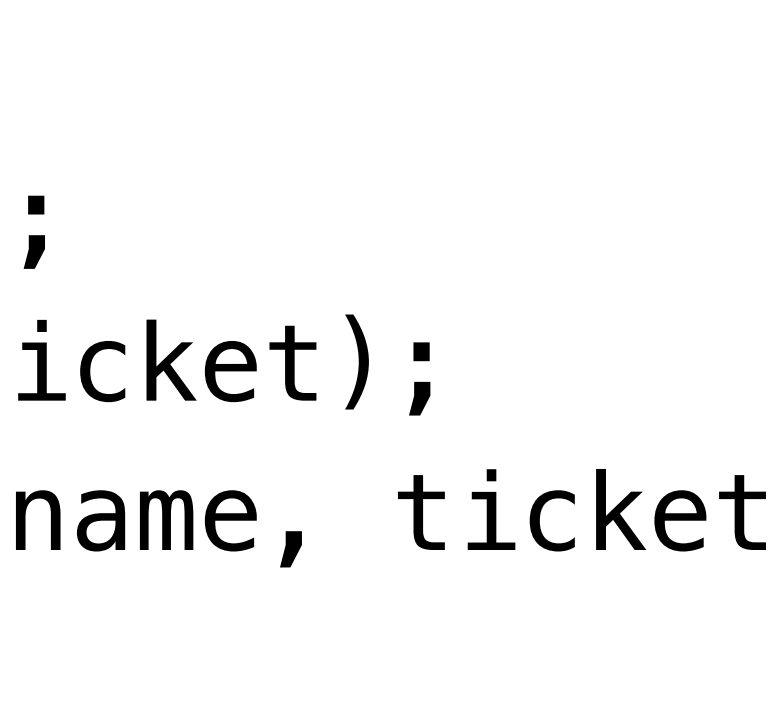
```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```



Workflow Method **Replay #1**

@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

A diagram consisting of a yellow rounded rectangle labeled "Skip" in the top right. A yellow arrow originates from the bottom-left corner of this box and points diagonally down and to the left, terminating at the line of code `sendConfirmationEmail(email, ticket, hotel, train);` in the code block below.

Workflow Method **Replay #2**

@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #2**

Get History

@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #2**

Get Result



@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #2**

Schedule



@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #2**

Schedule



@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```

Workflow Method **Replay #2**

@Override

```
public void prepareForJEEConf(String name, String email) {  
    Promise<Integer> ticket = jeeConfService.buyTicket(name);  
    Promise<Integer> hotel = travelService.bookHotel(name, ticket);  
    Promise<Integer> train = travelService.bookTrainTickets(name, ticket);  
  
    sendConfirmationEmail(email, ticket, hotel, train);  
}
```



Skip

Activity Timeouts

Activity **Heartbeats**



History and Data Size **Limits**



25K

History events



32K

Message size
(chars)

Summary



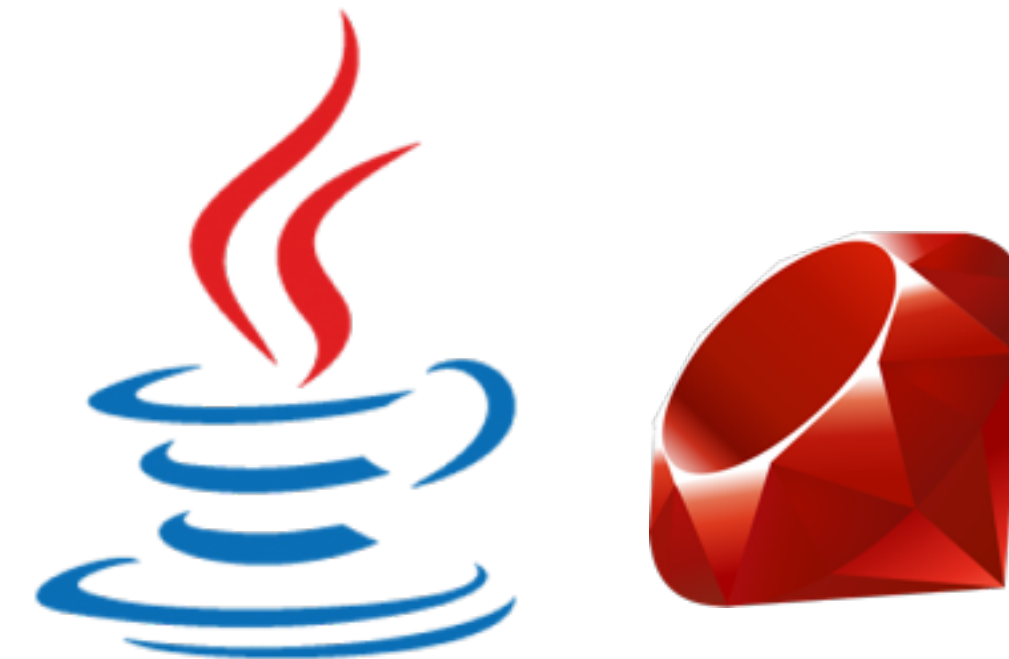
Programs with steps



Scalability



Timeouts & Retries



Java & Ruby



Questions

Serhiy Batyuk

✉ sbatyuk@gmail.com

🐦 [@sbatyuk](https://twitter.com/sbatyuk)