Simple Workflow

Distributed Out of the Box!

Serhiy Batyuk
JEEConf
May 21, 2016

Tech Lead at LOHIKA

AWS SWF > **1 year**

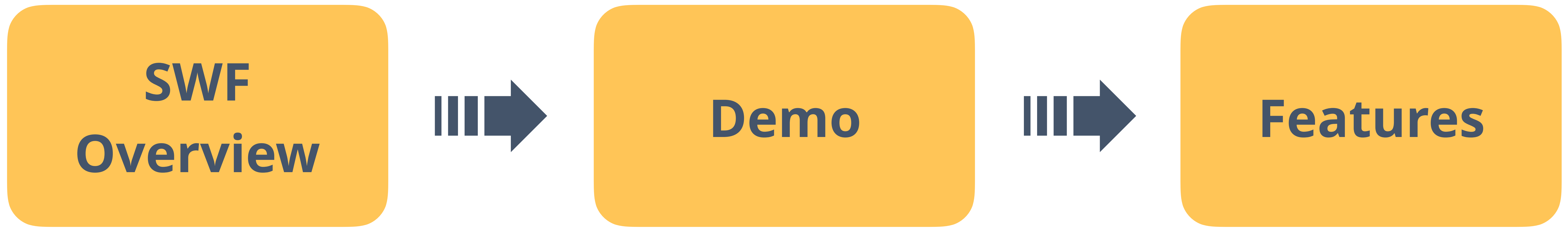
Agenda

SWF Overview

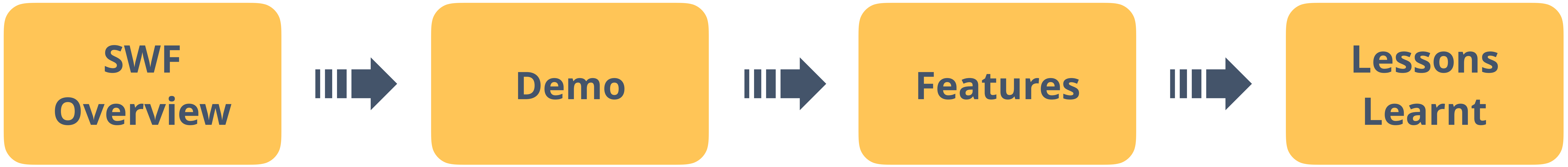
Agenda

SWF Overview → Demo

Agenda



Agenda



What Is **AWS Simple Workflow**?

AWS SWF **Use Cases**

Media Processing

Infrastructure Provisioning

Data Processing

Report Generation

Business Processes

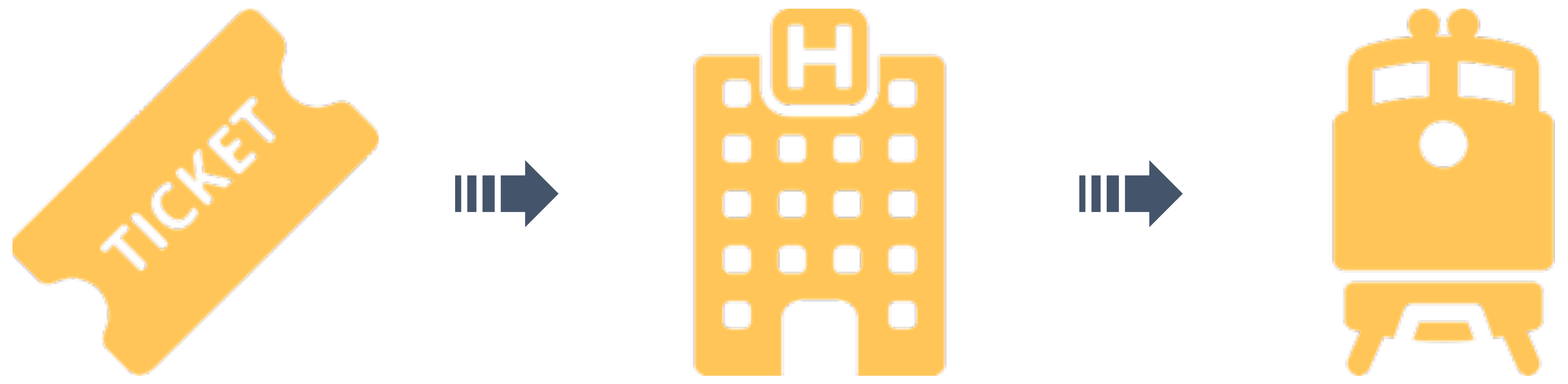
https://www.youtube.com/watch?v=DYmJIQO2ZyQ&t=83

https://globallogic.com.ua/speaking-event/jeeconf-2015

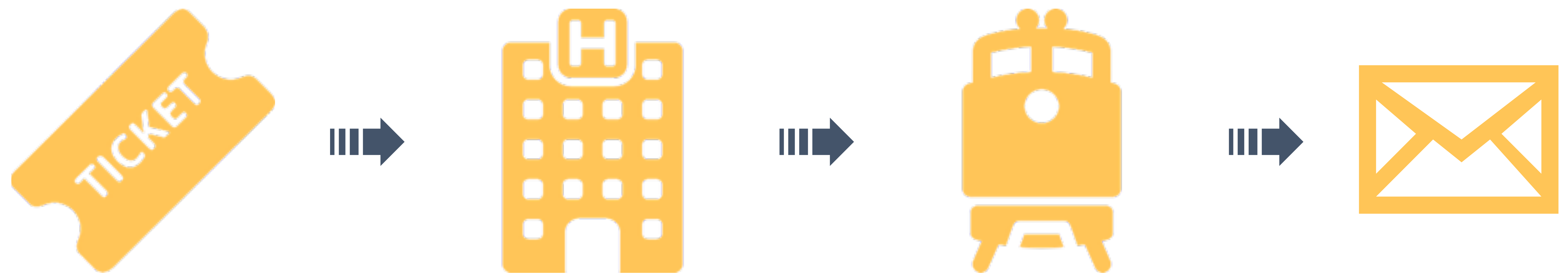
Preparing to Attend **JEEConf**

Preparing to Attend **JEEConf**

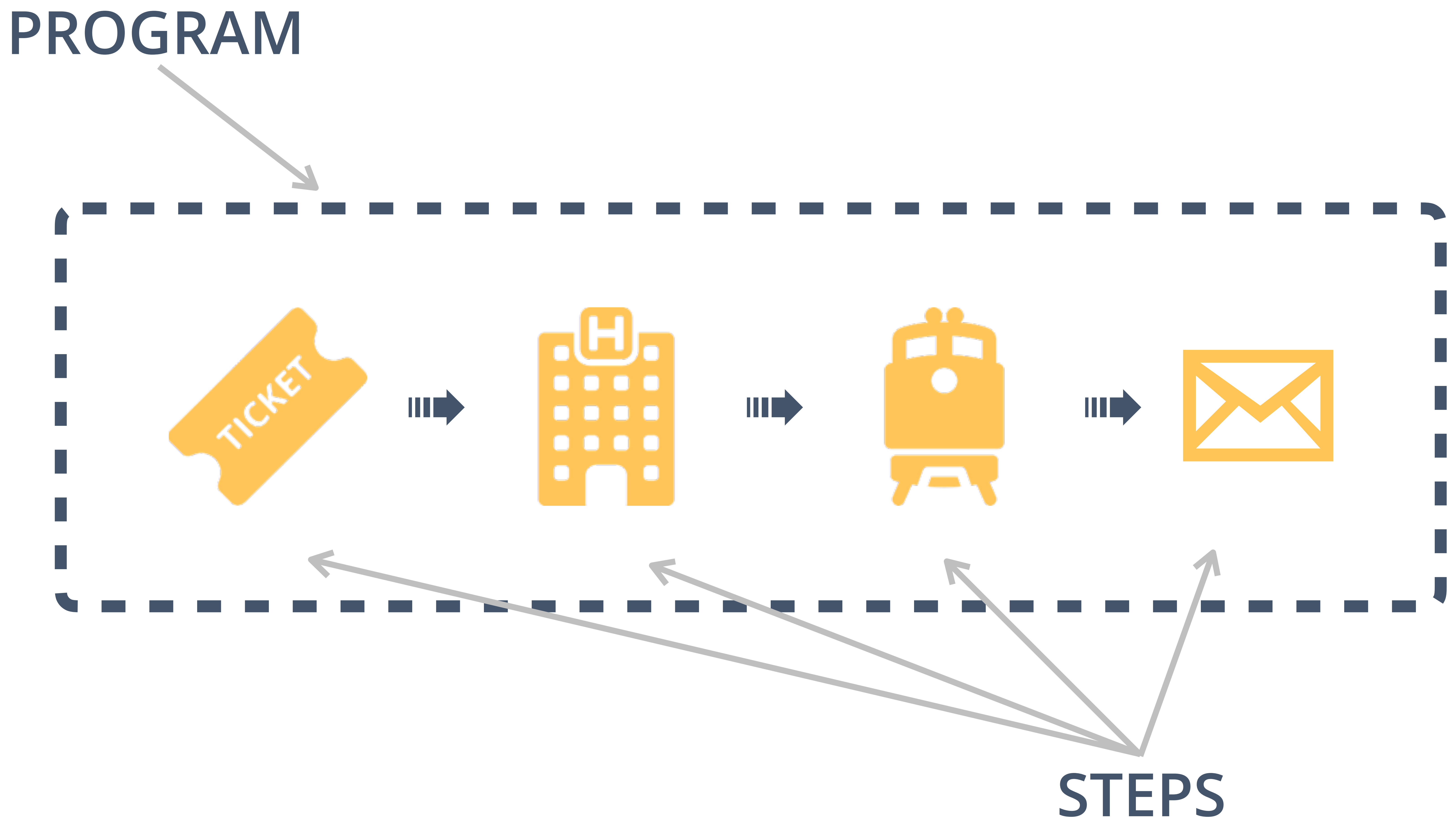
Preparing to Attend **JEEConf**



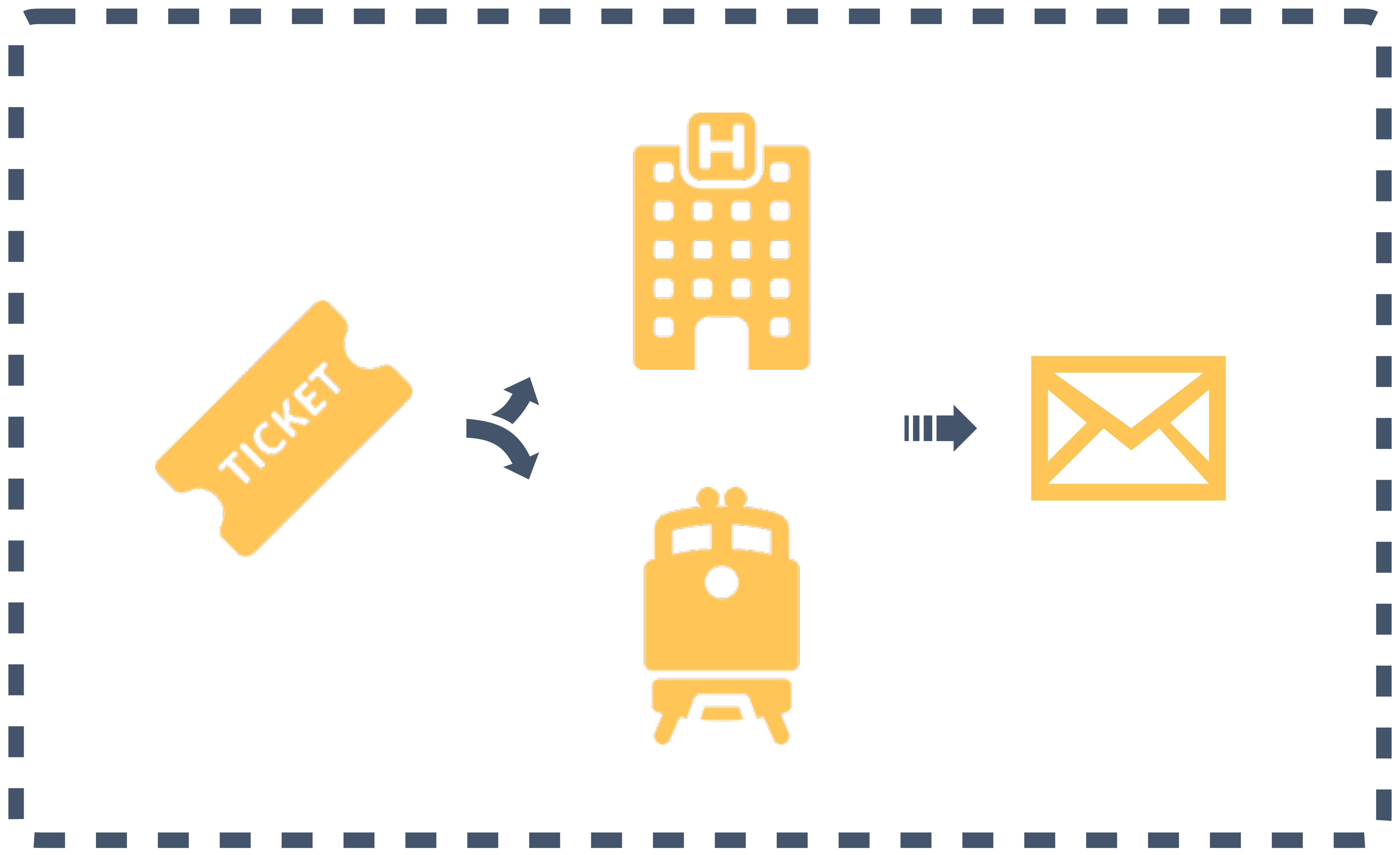
Preparing to Attend **JEEConf**



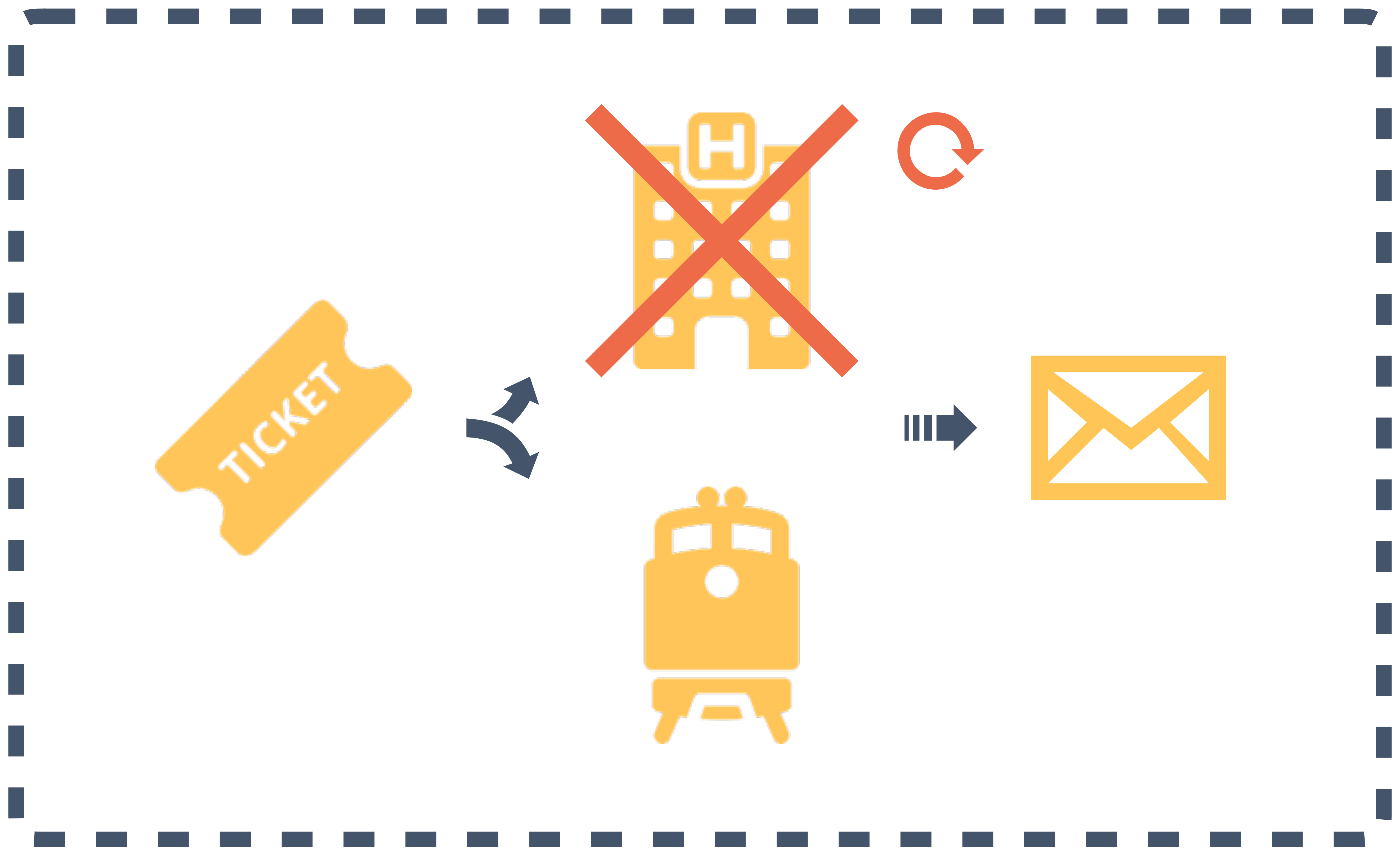
Preparing to Attend **JEEConf**



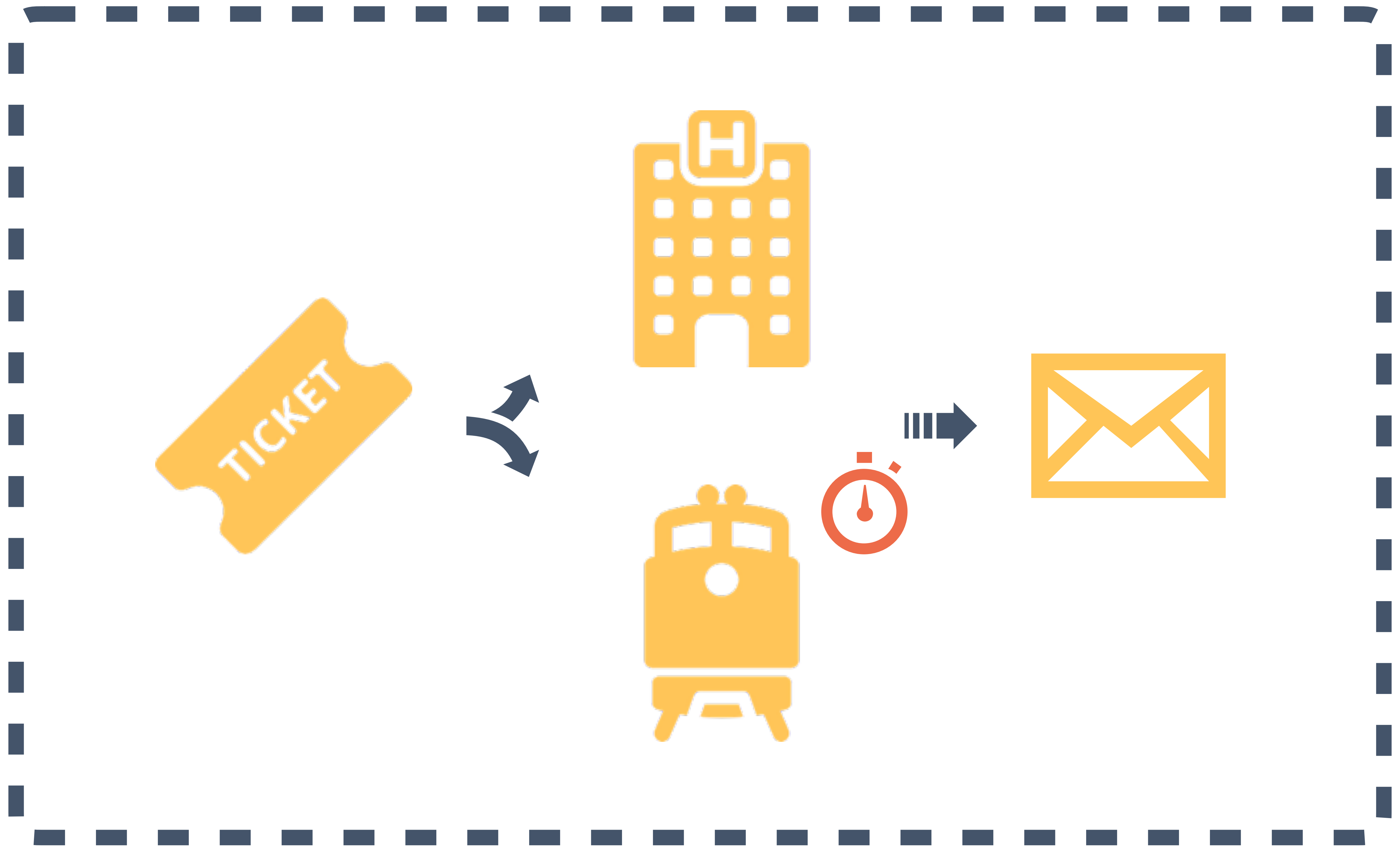
Preparing to Attend **JEEConf**



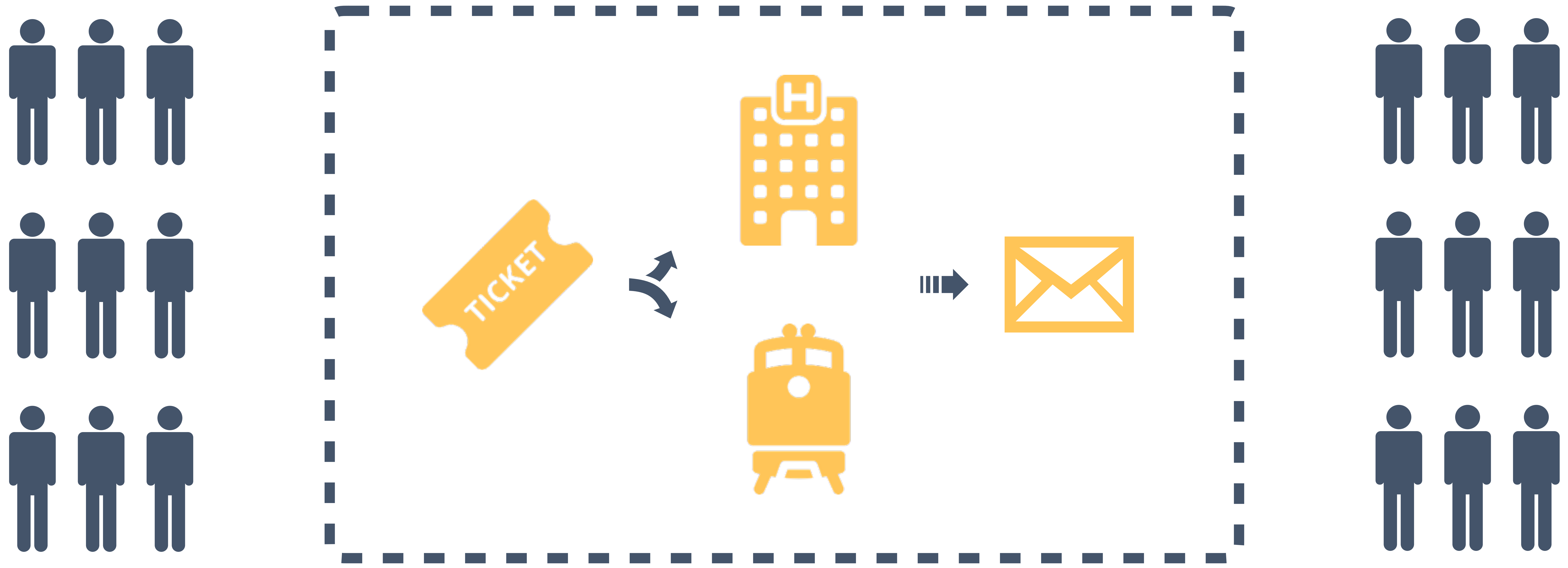
Preparing to Attend **JEEConf**



Preparing to Attend **JEEConf**



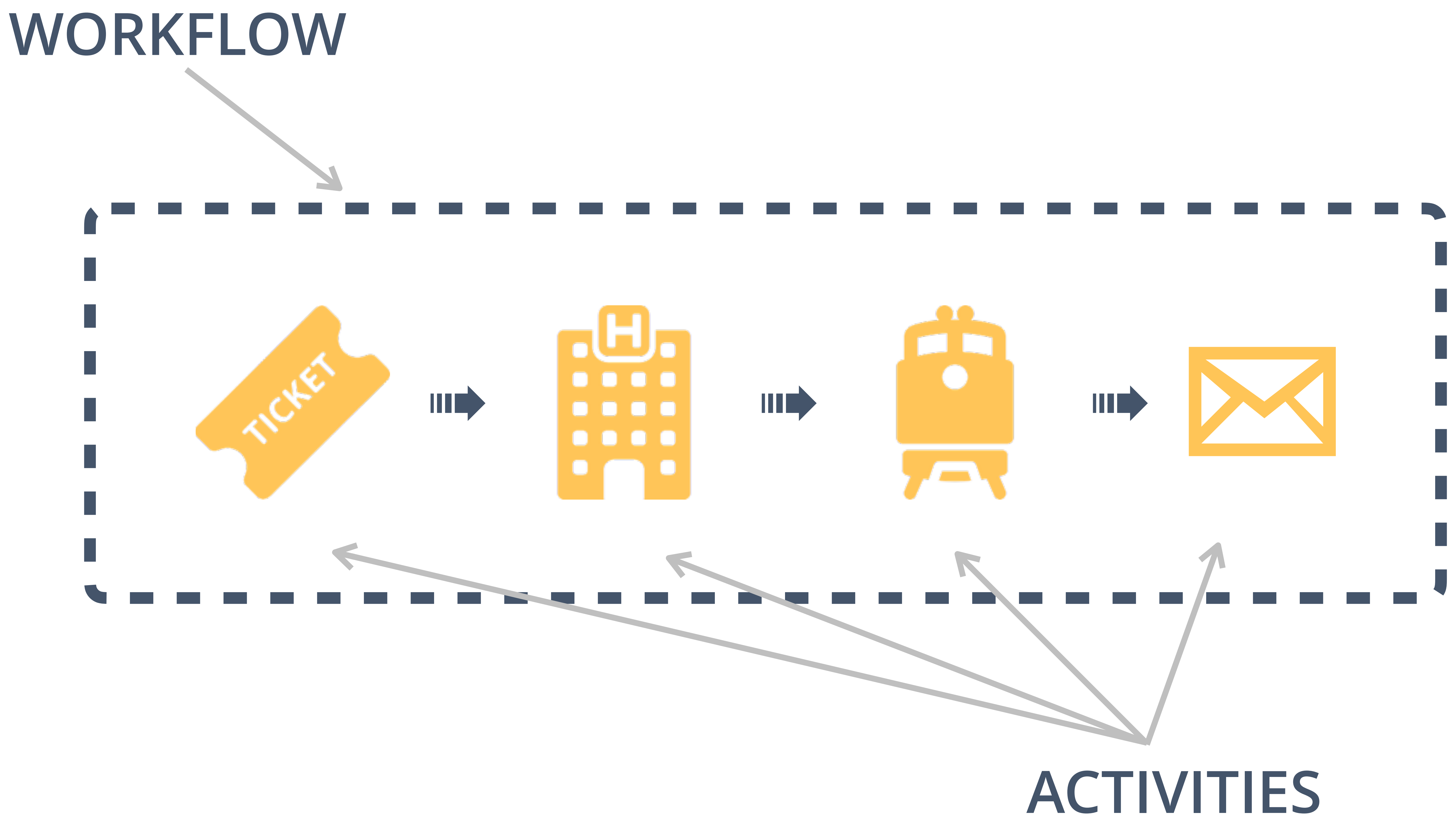
Preparing to Attend **JEEConf**



AWS SWF

https://aws.amazon.com/swf

Workflows and **Activities**

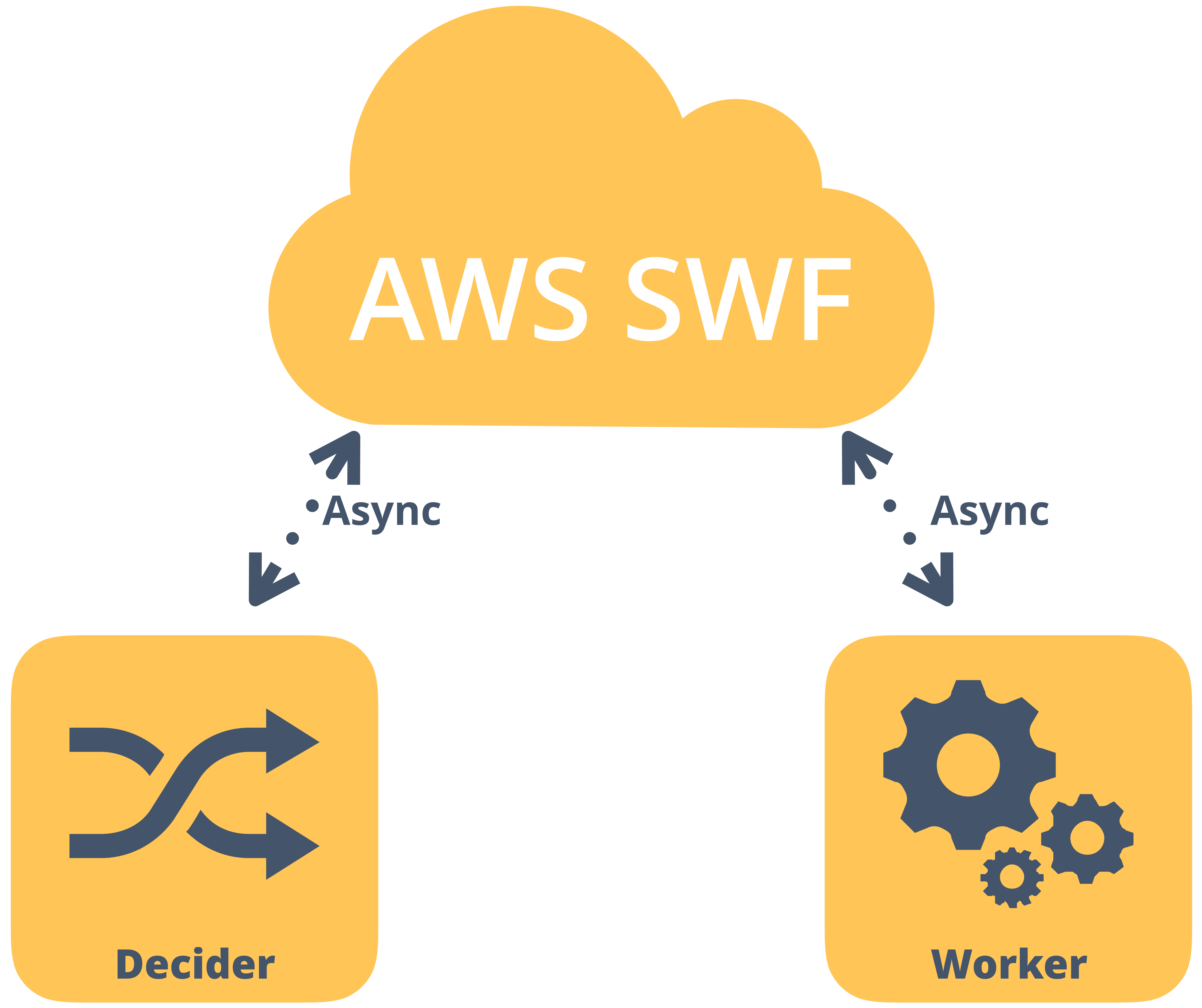


SWF Application **Architecture**

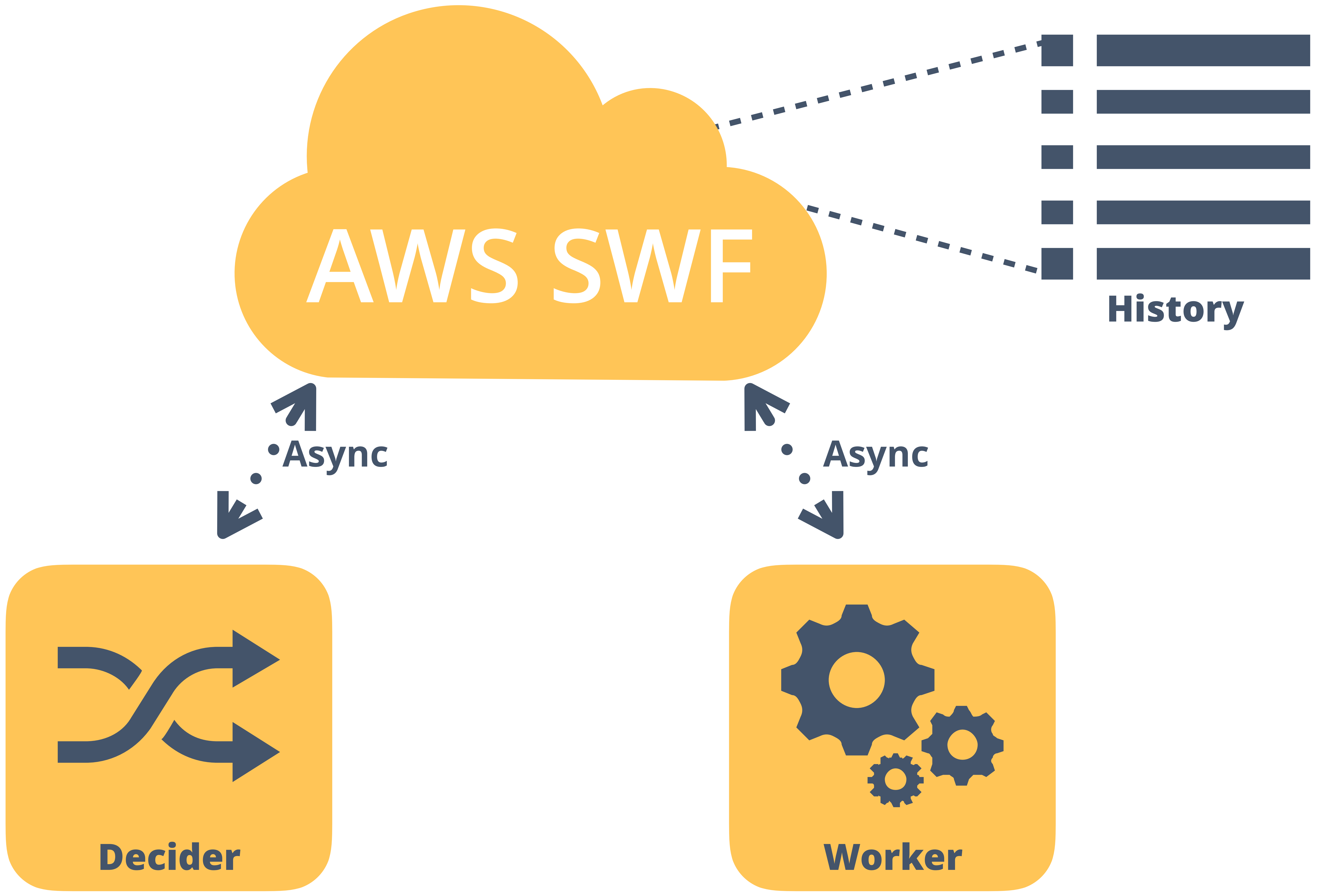
Decider

Worker

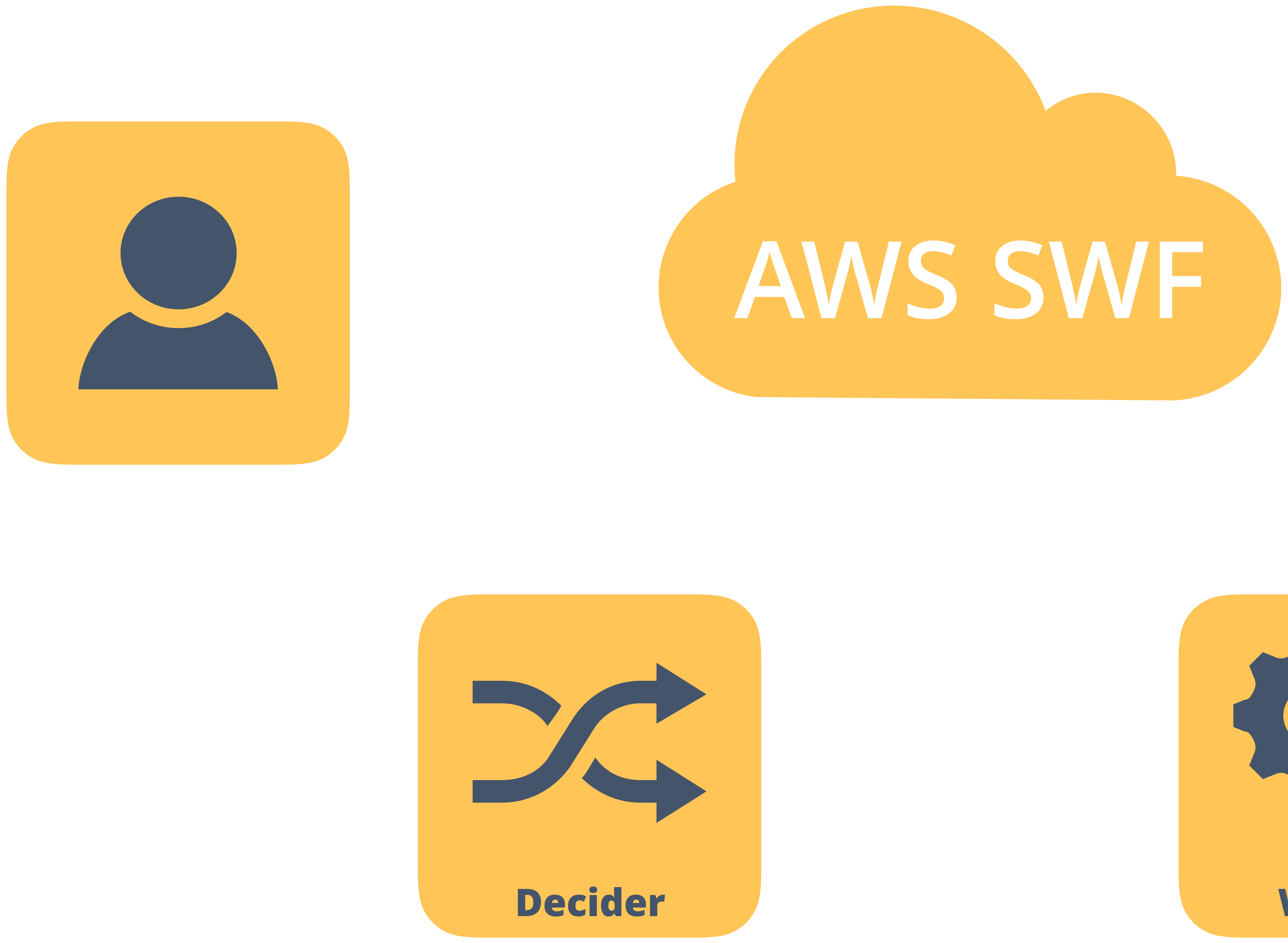
SWF Application **Architecture**



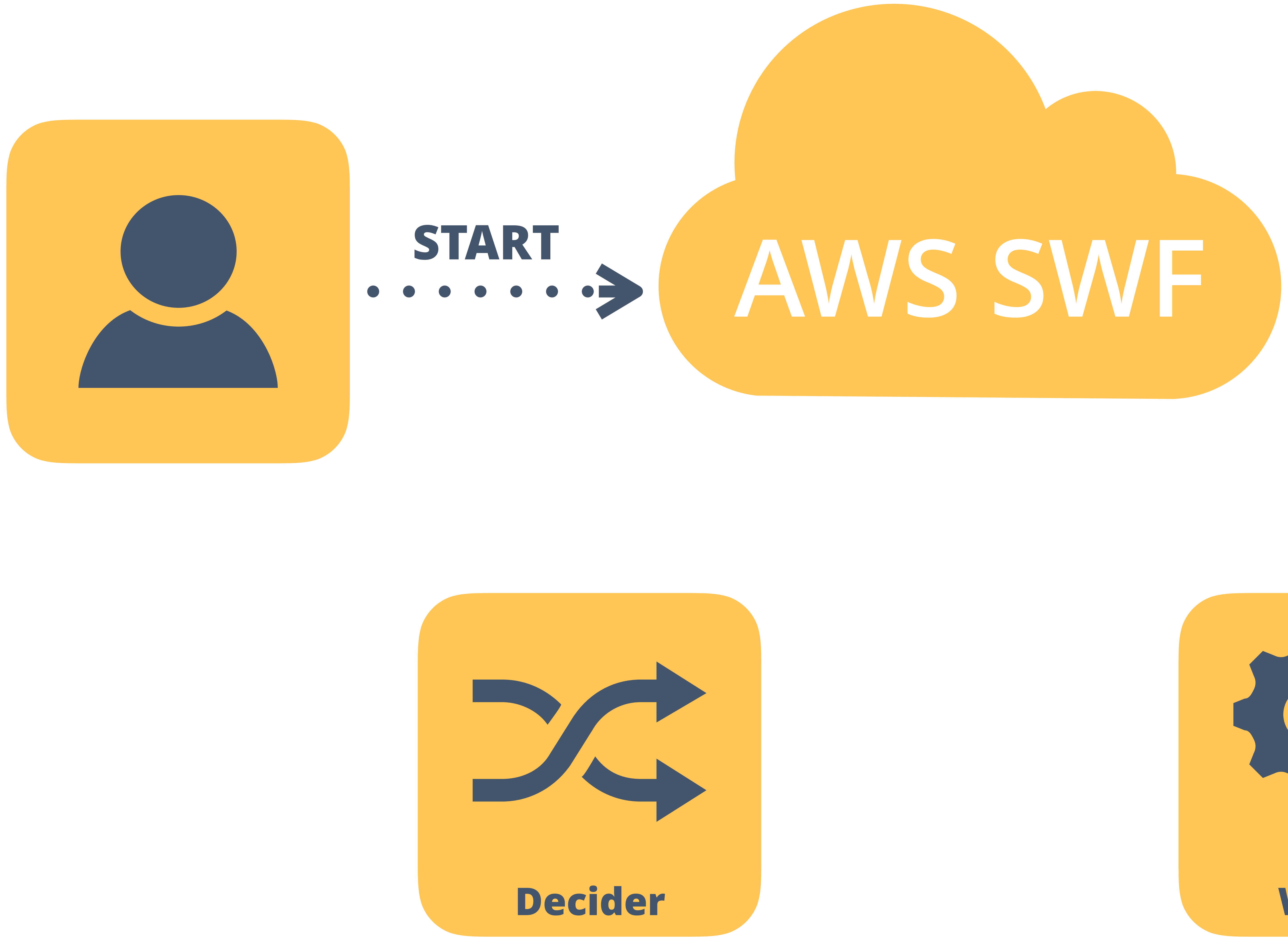
SWF Application **Architecture**



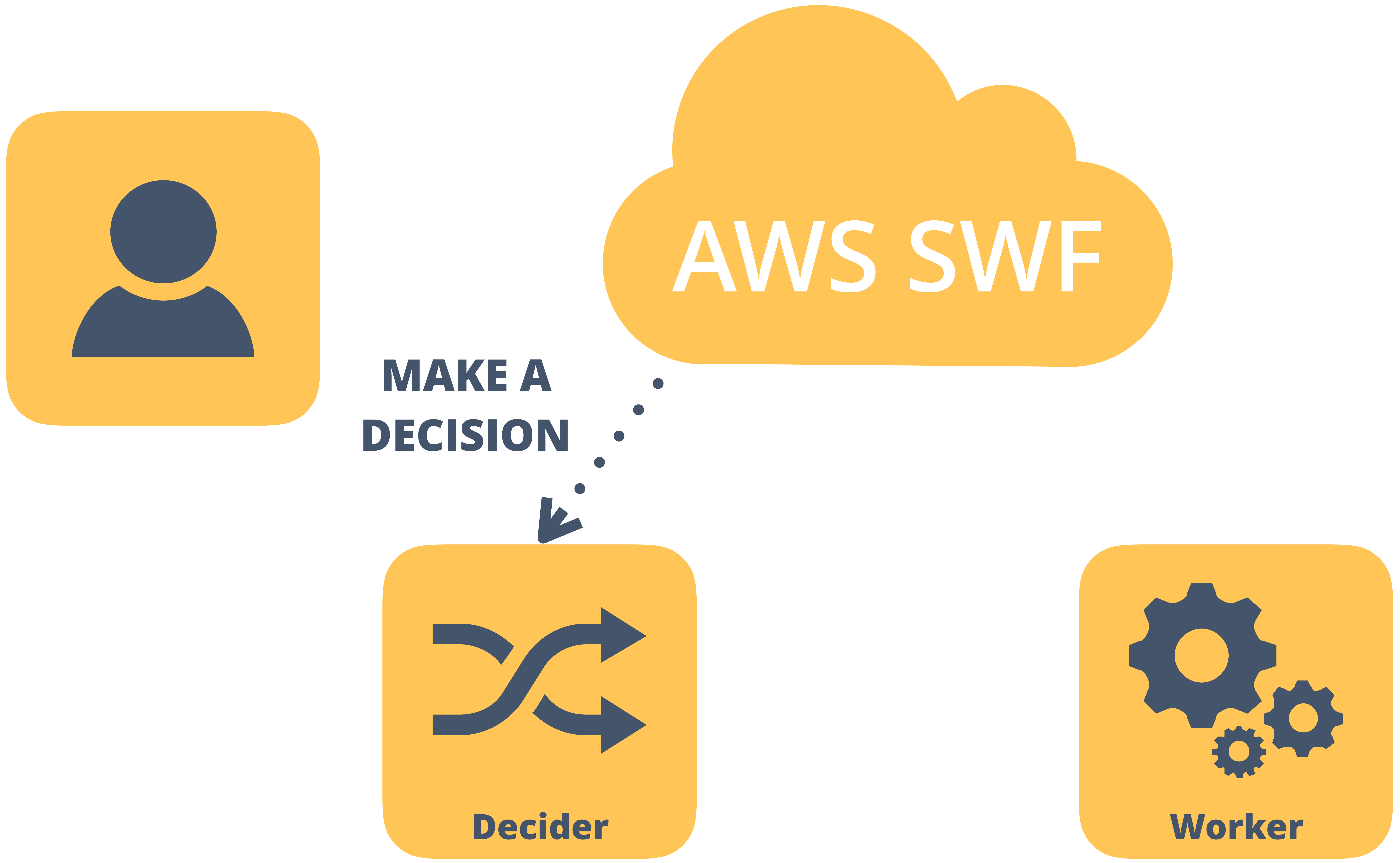
AWS SWF **Message Flow**



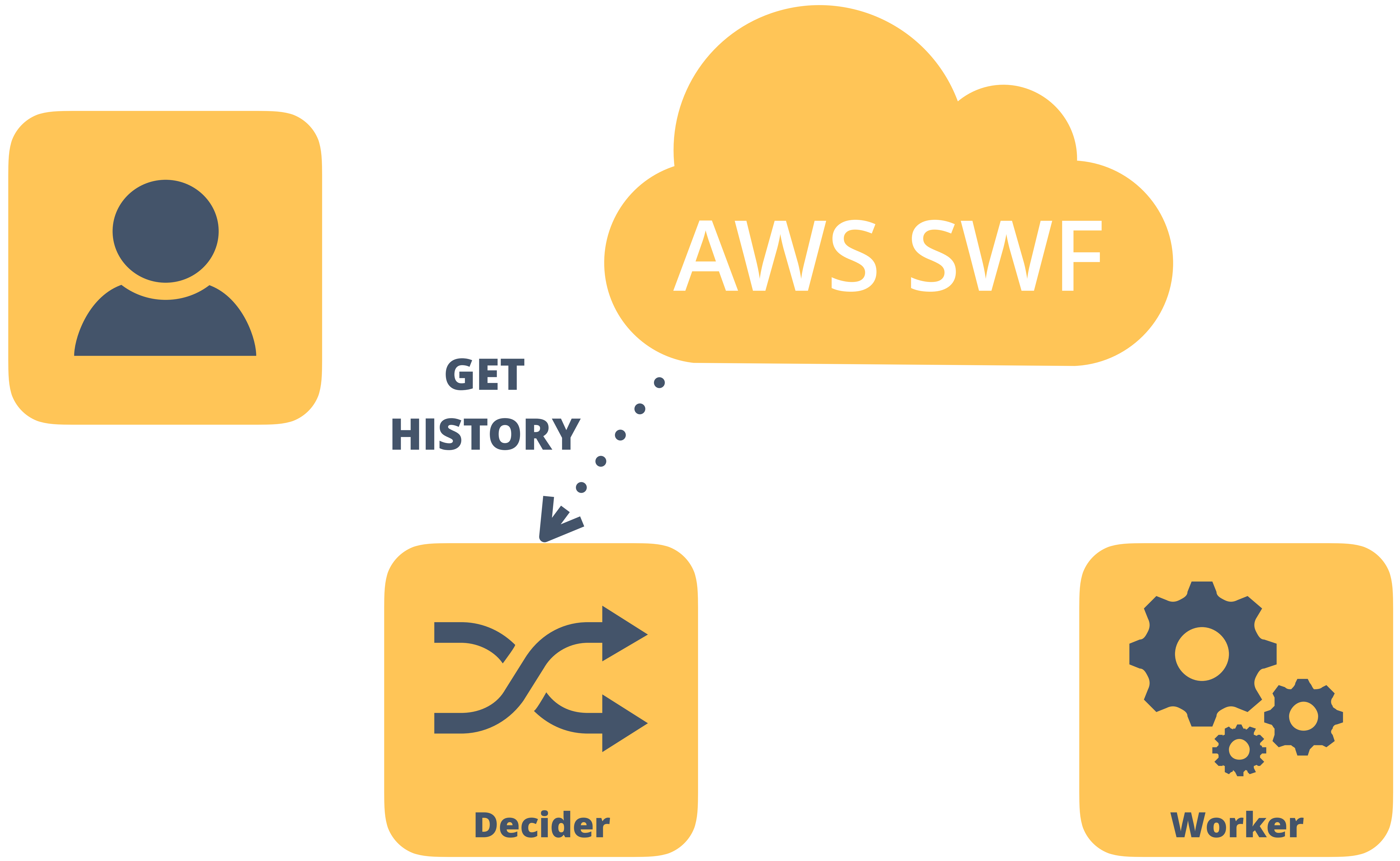
AWS SWF **Message Flow**



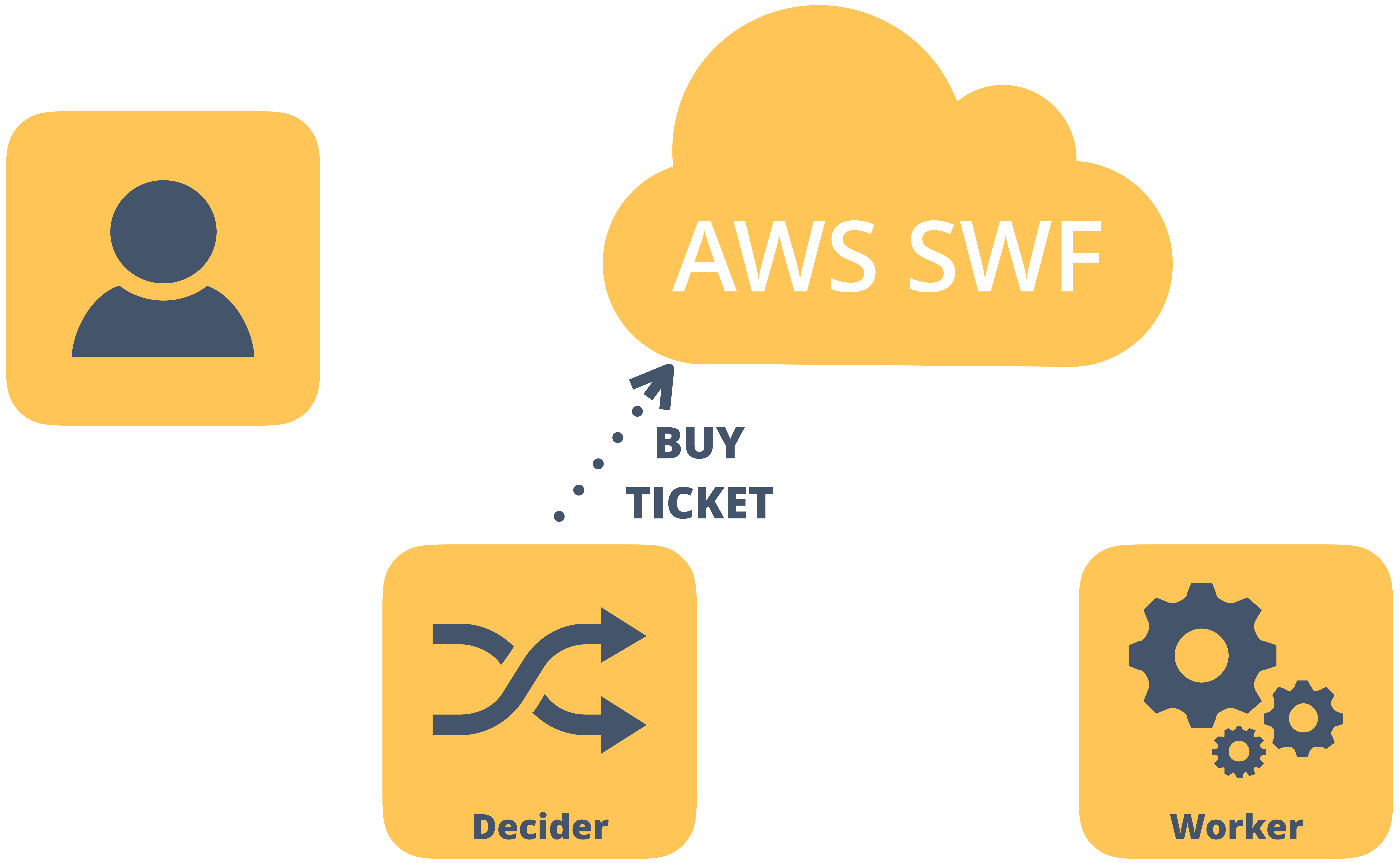
AWS SWF **Message Flow**



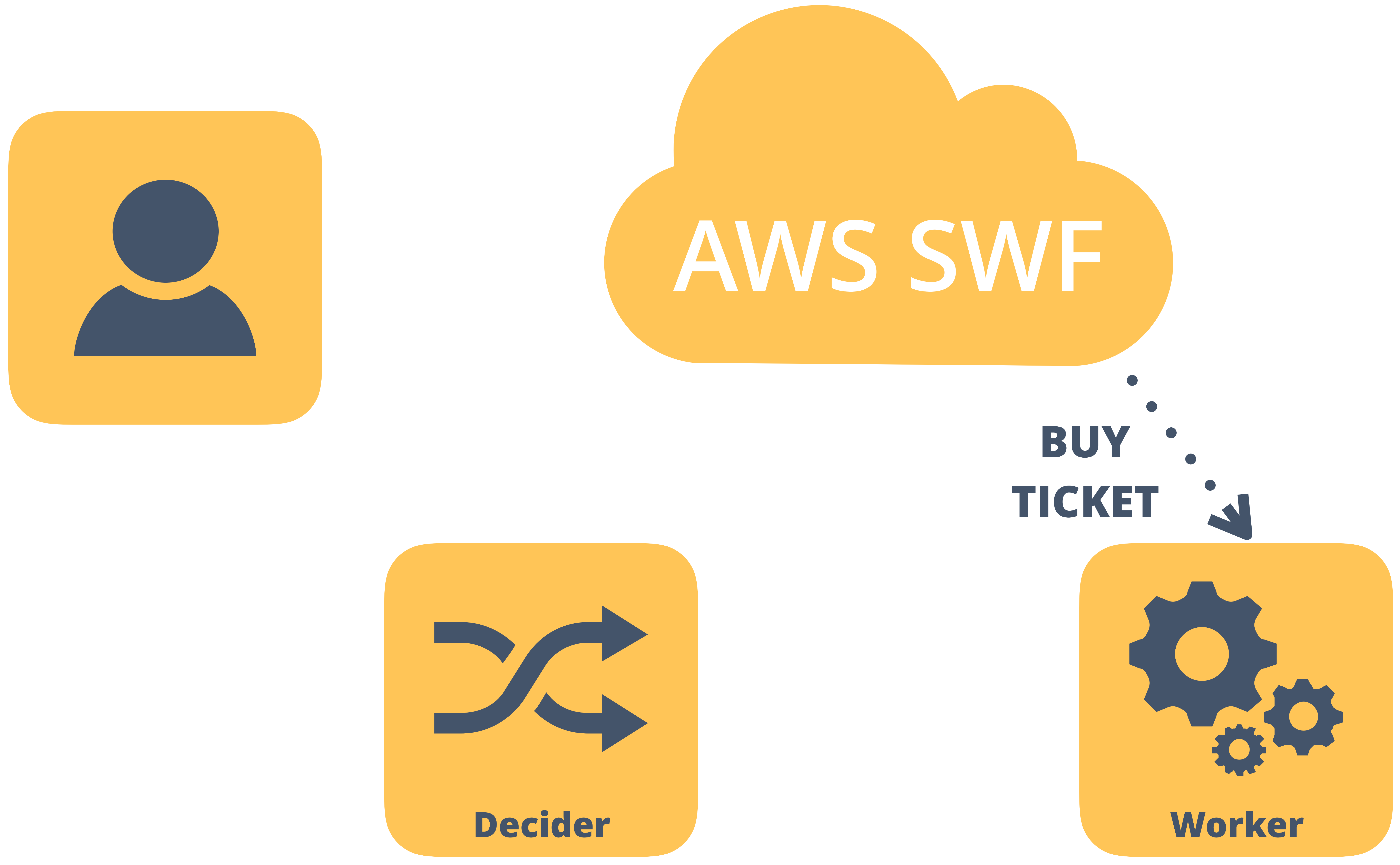
AWS SWF **Message Flow**



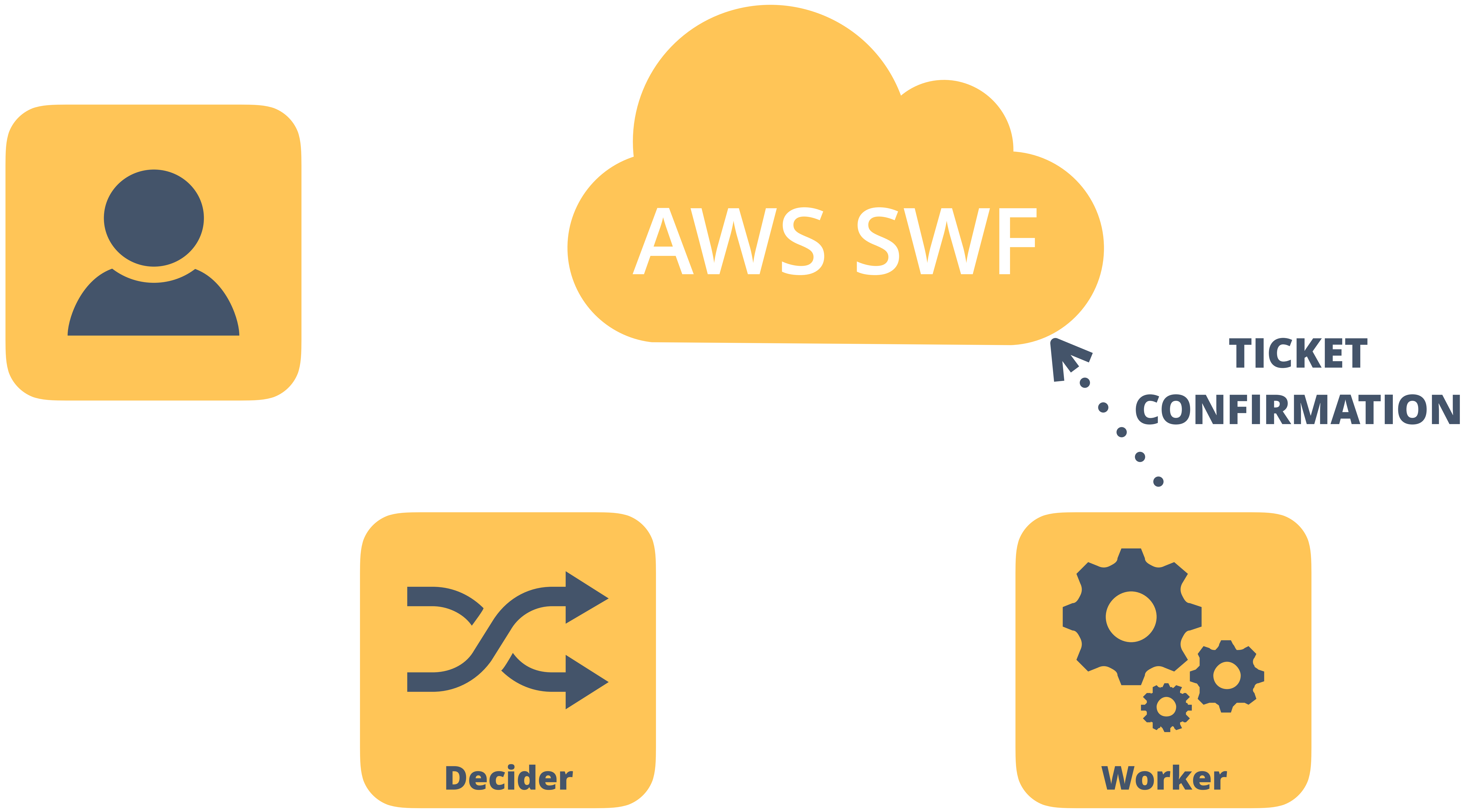
AWS SWF **Message Flow**



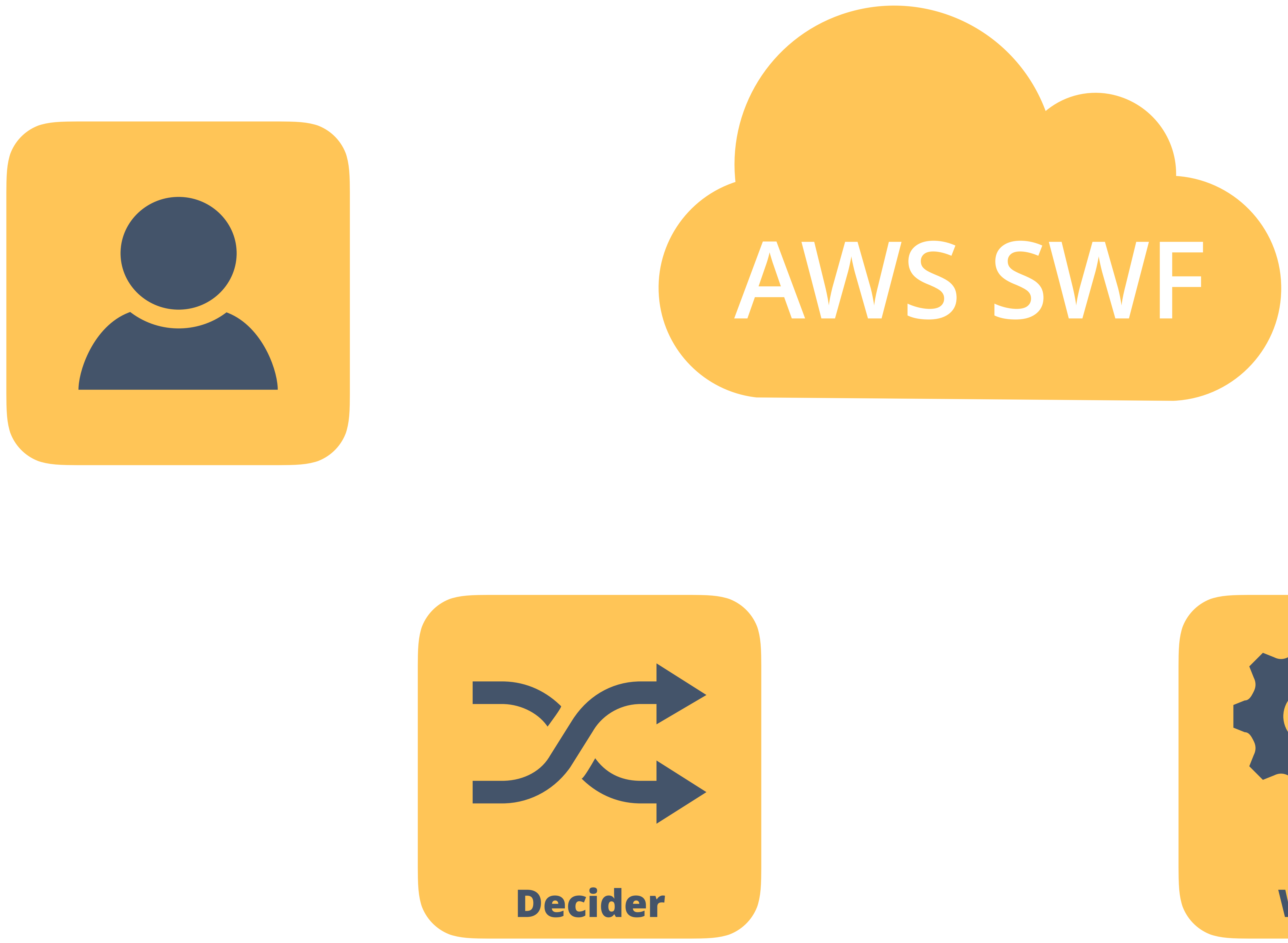
AWS SWF **Message Flow**



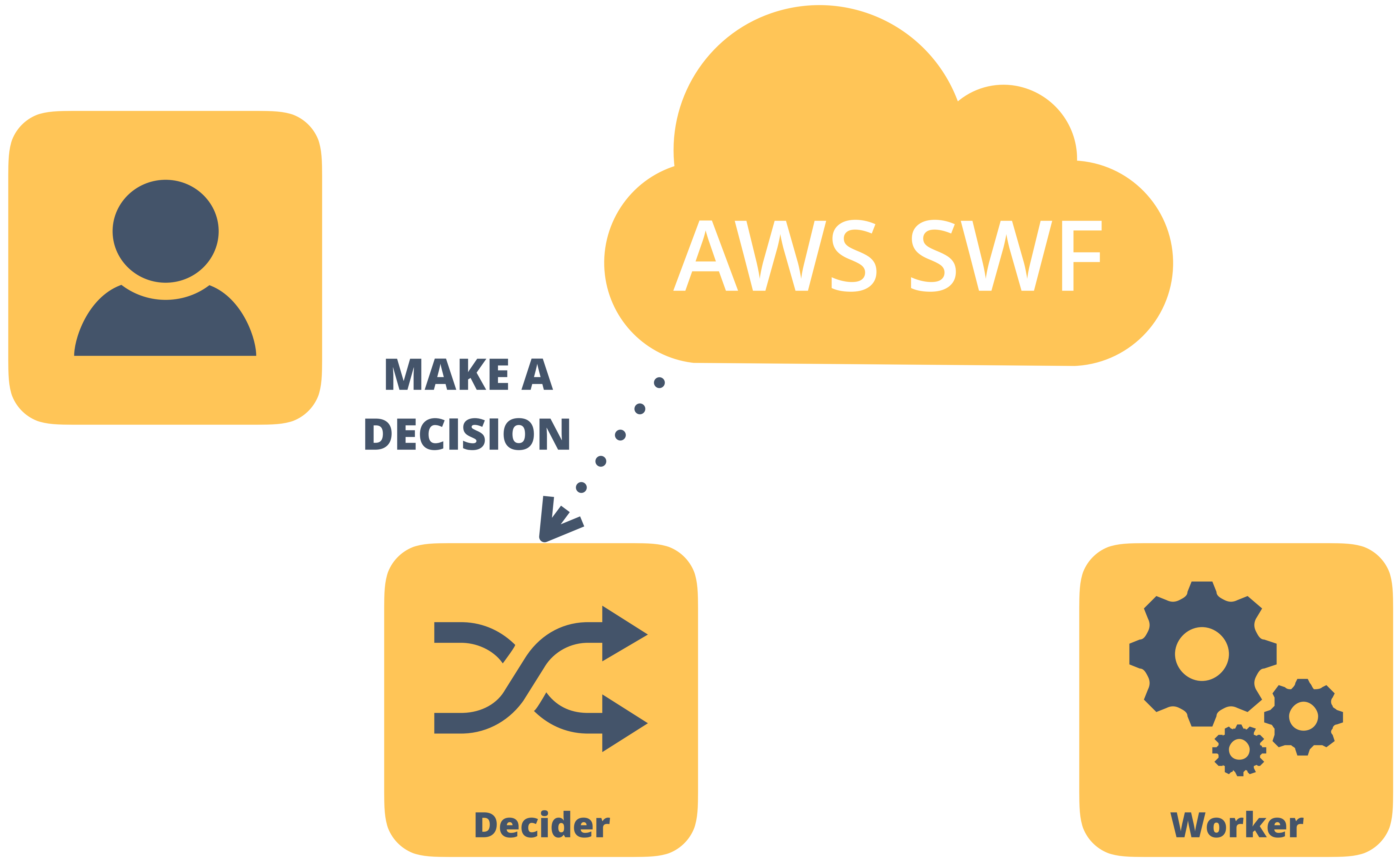
AWS SWF **Message Flow**



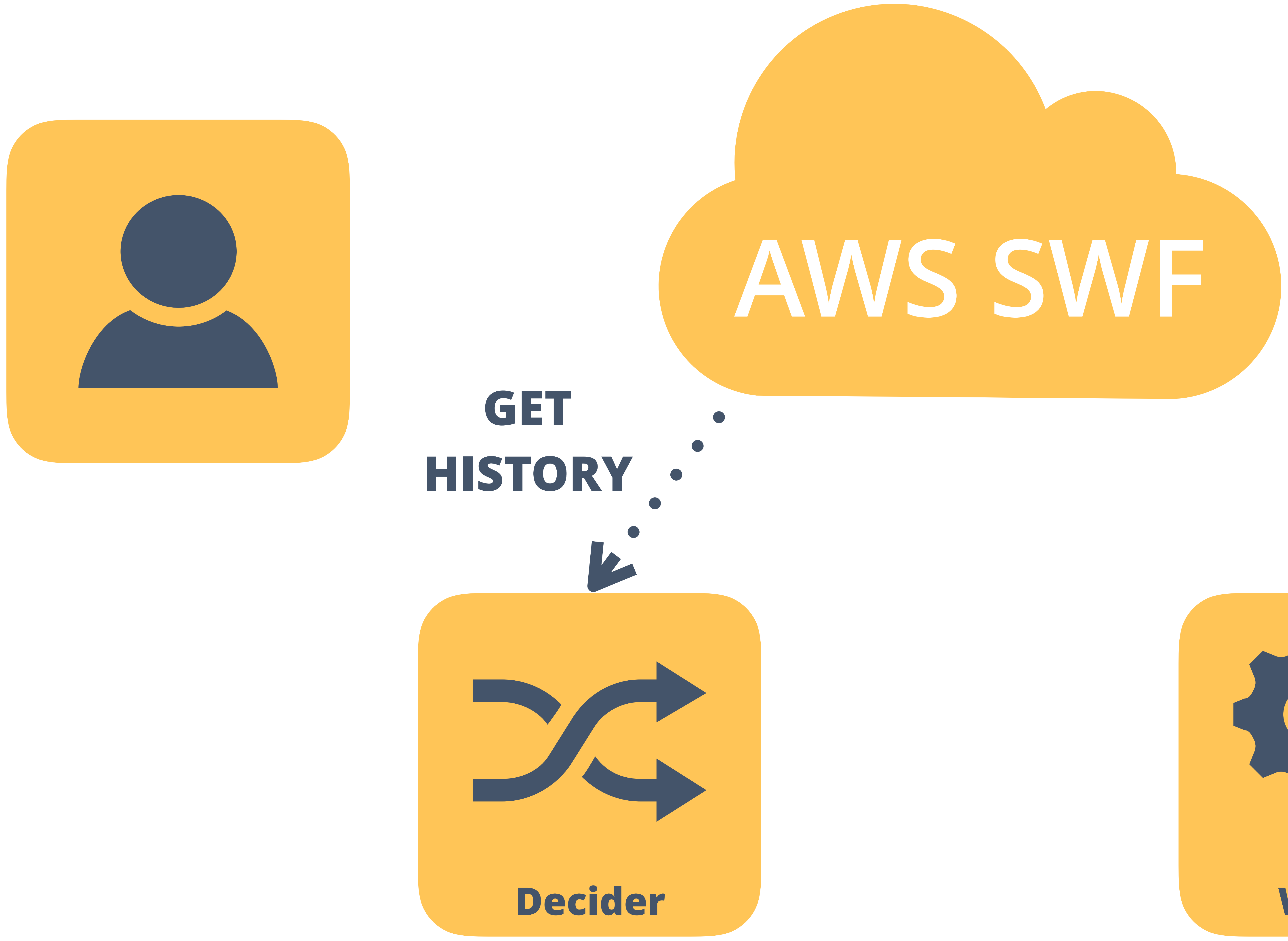
AWS SWF **Message Flow**



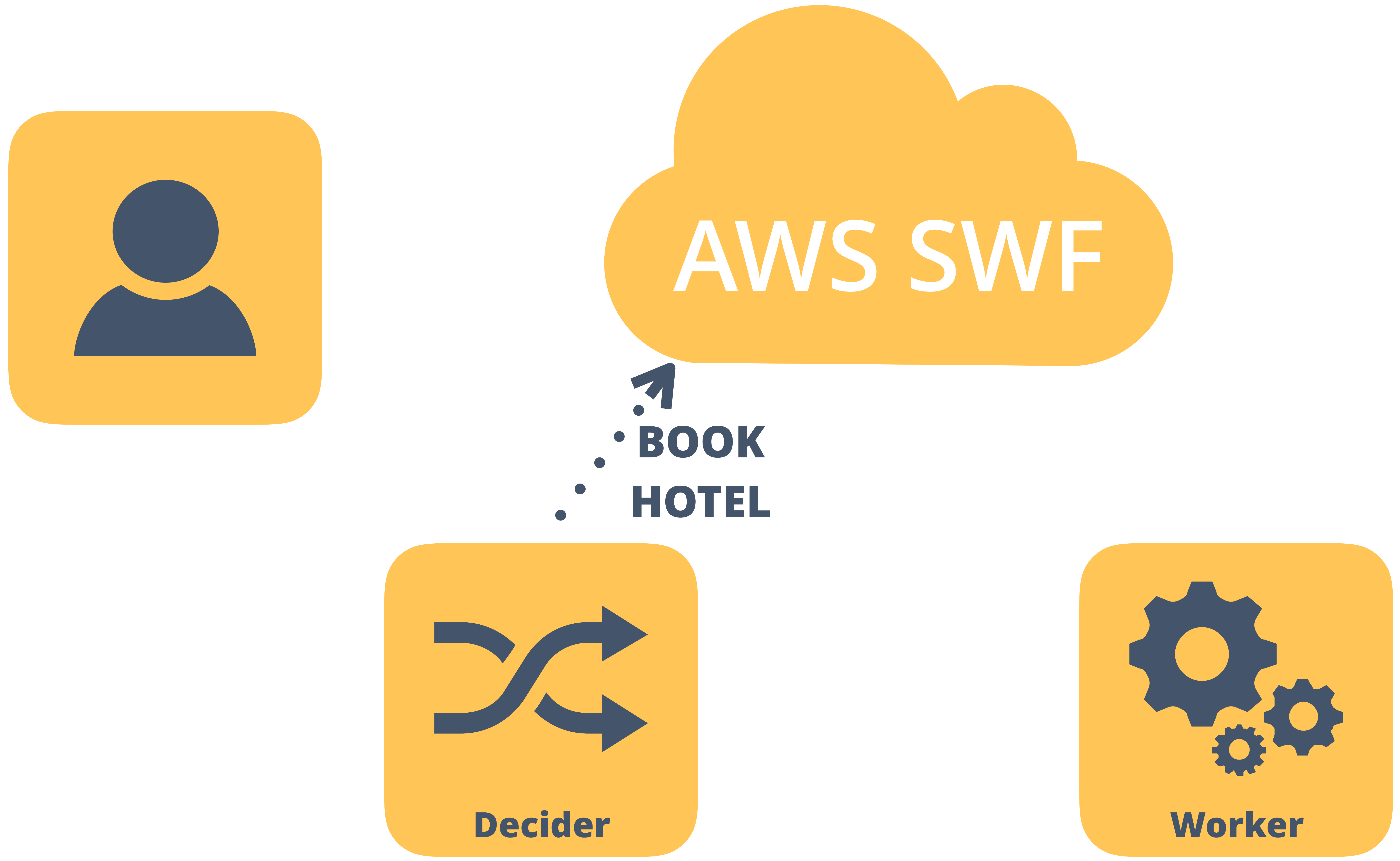
AWS SWF **Message Flow**



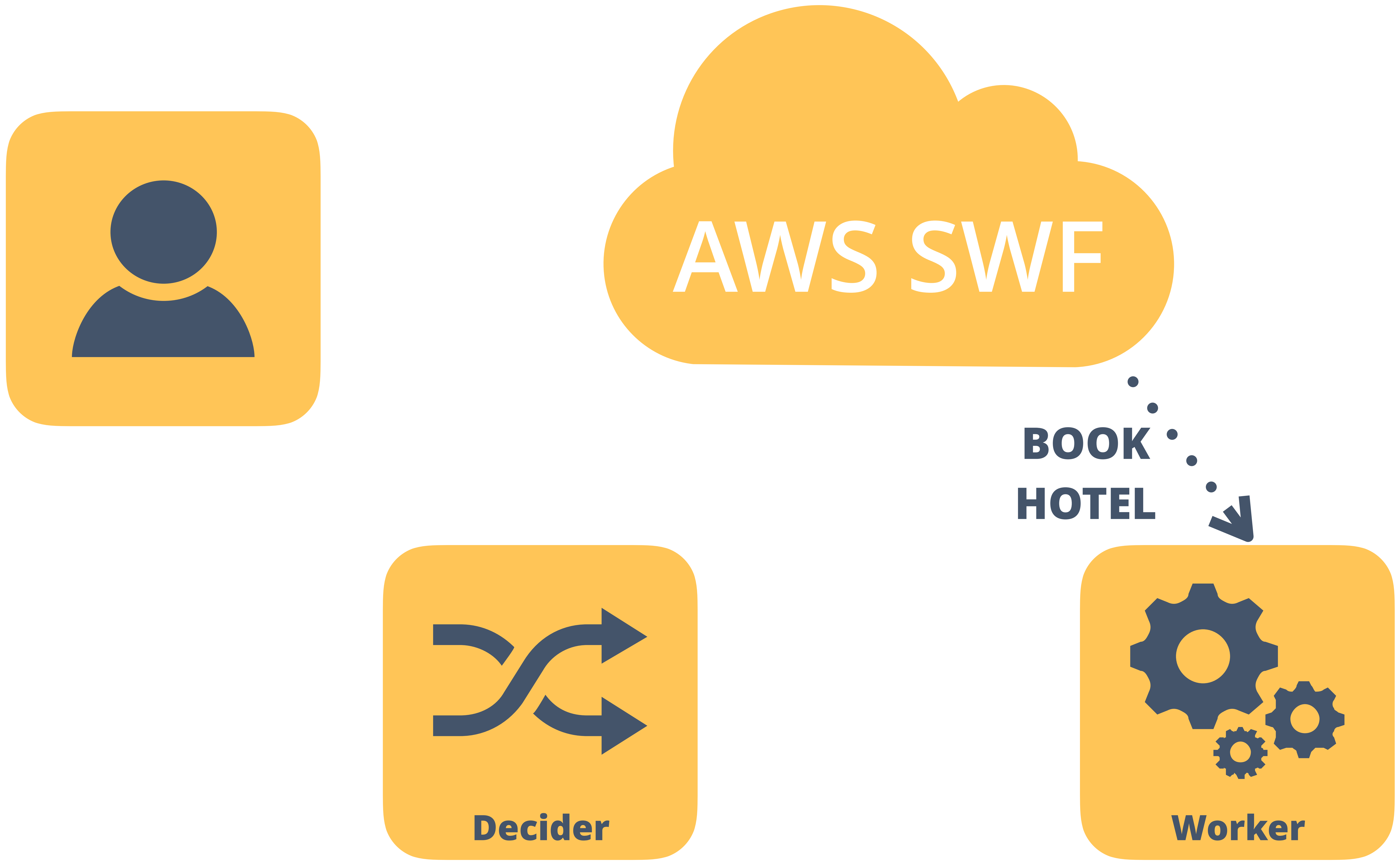
AWS SWF **Message Flow**



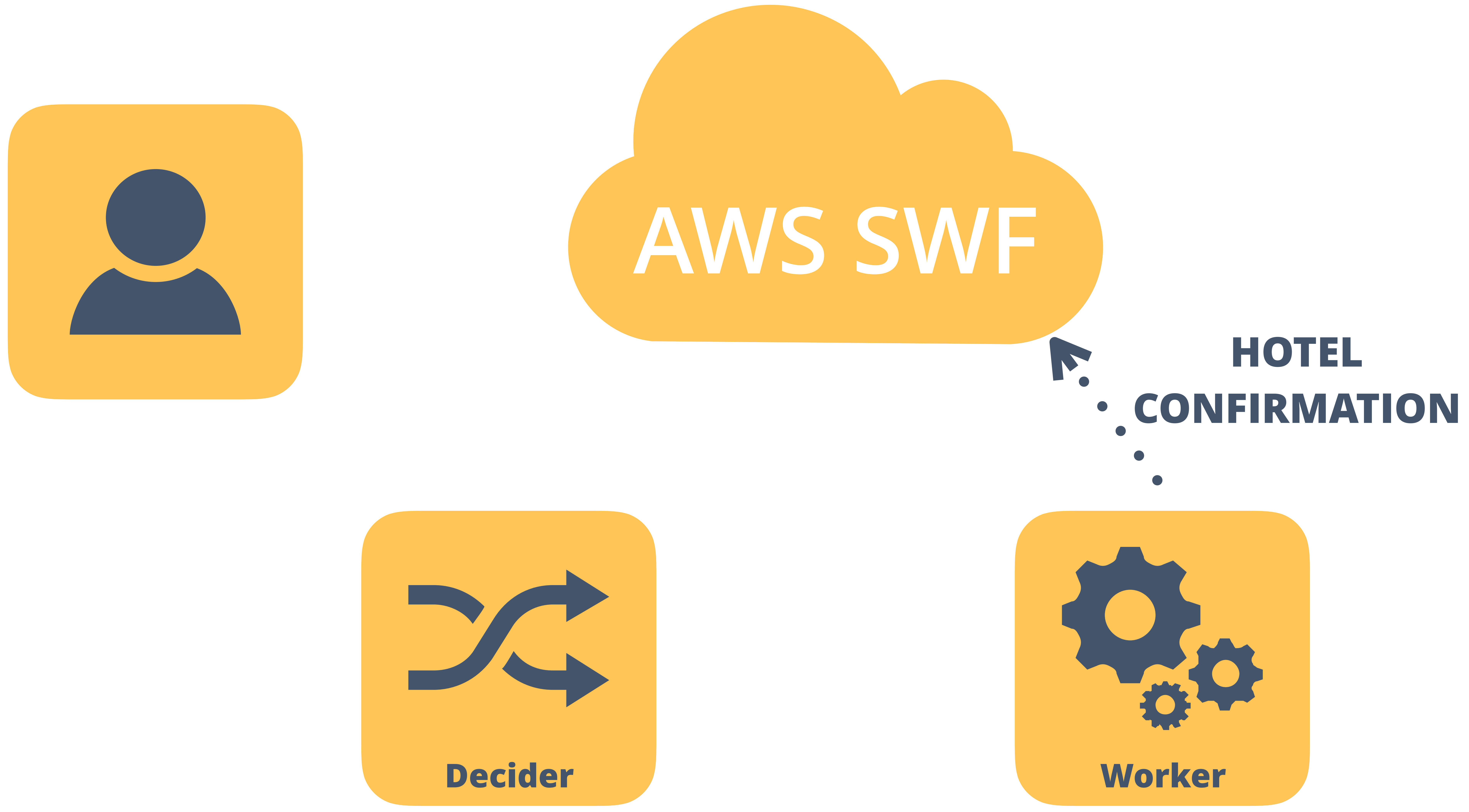
AWS SWF **Message Flow**



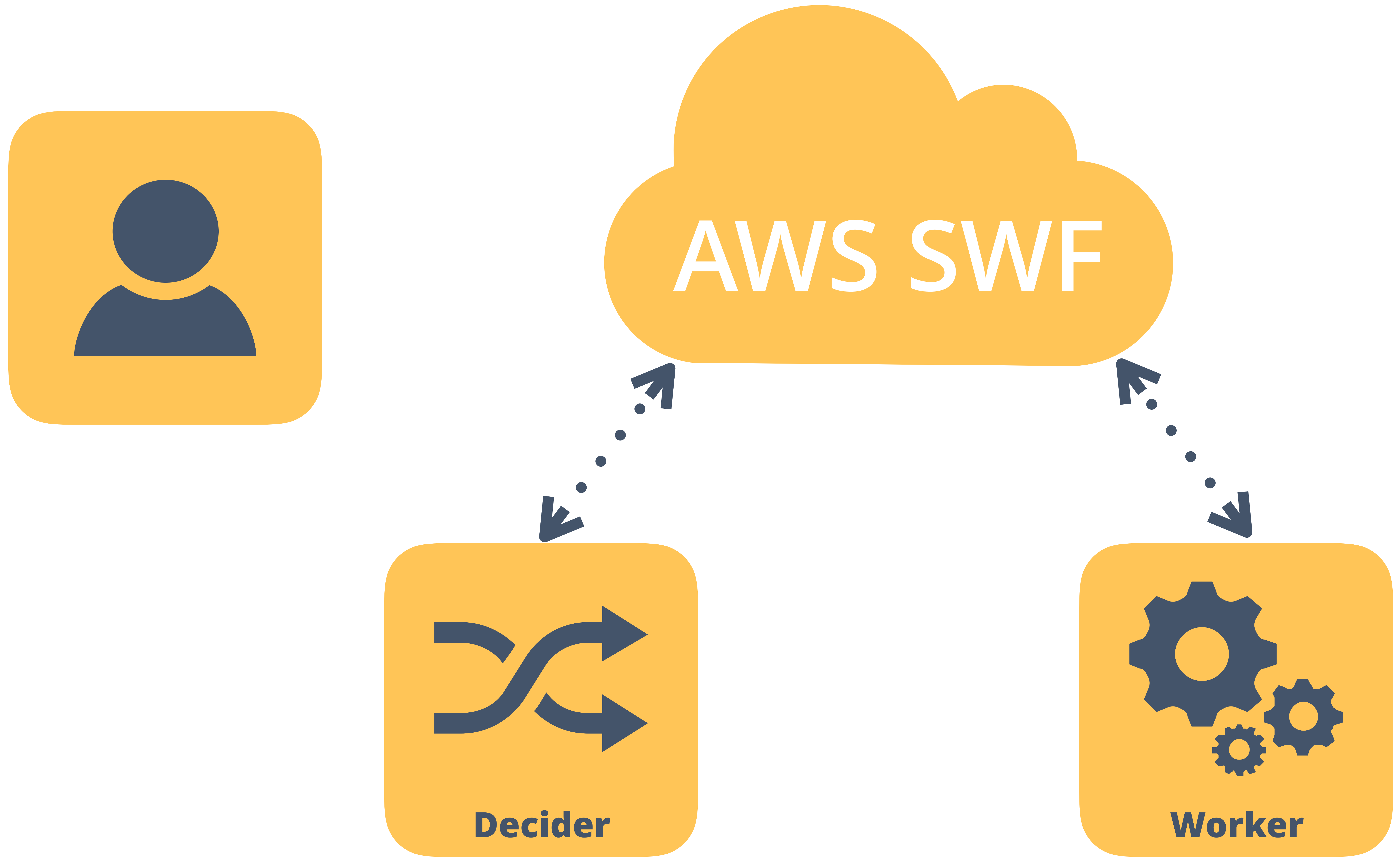
AWS SWF **Message Flow**



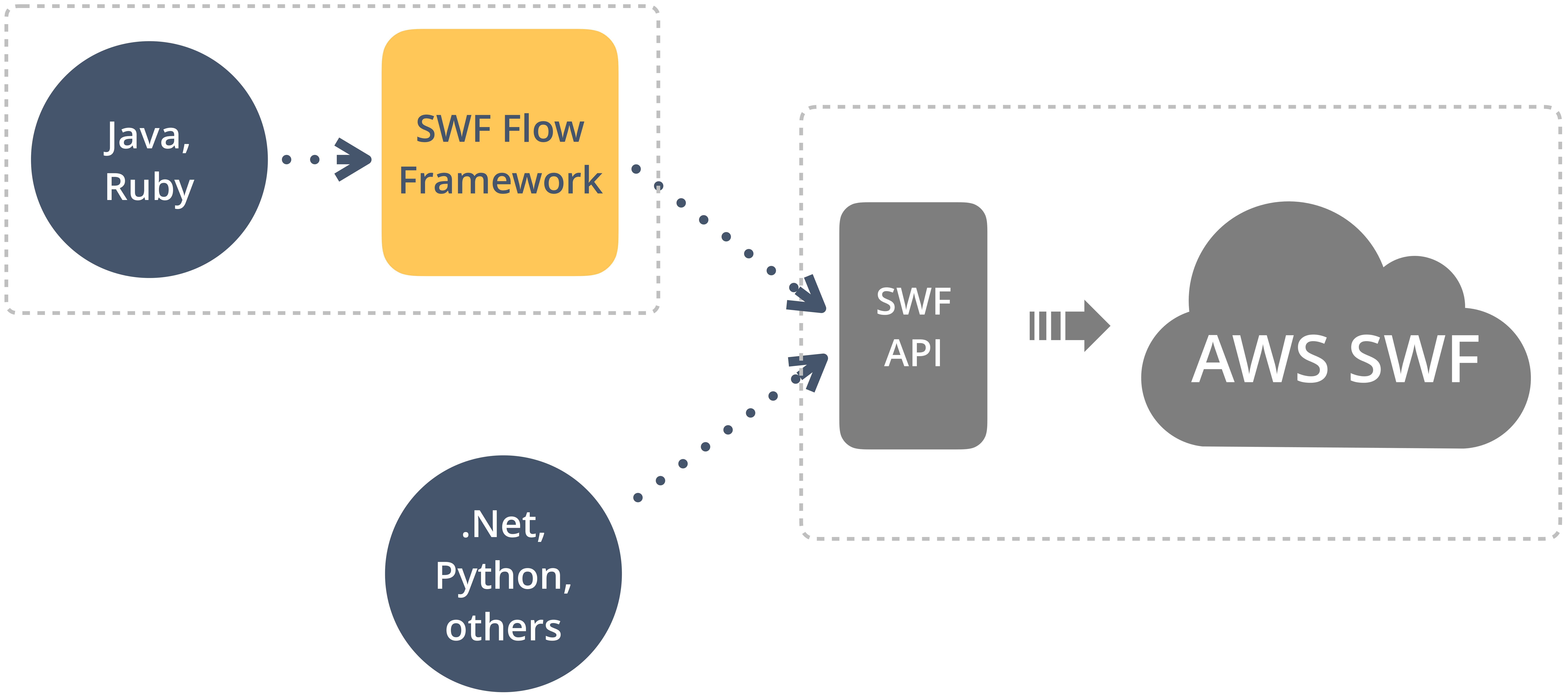
AWS SWF **Message Flow**



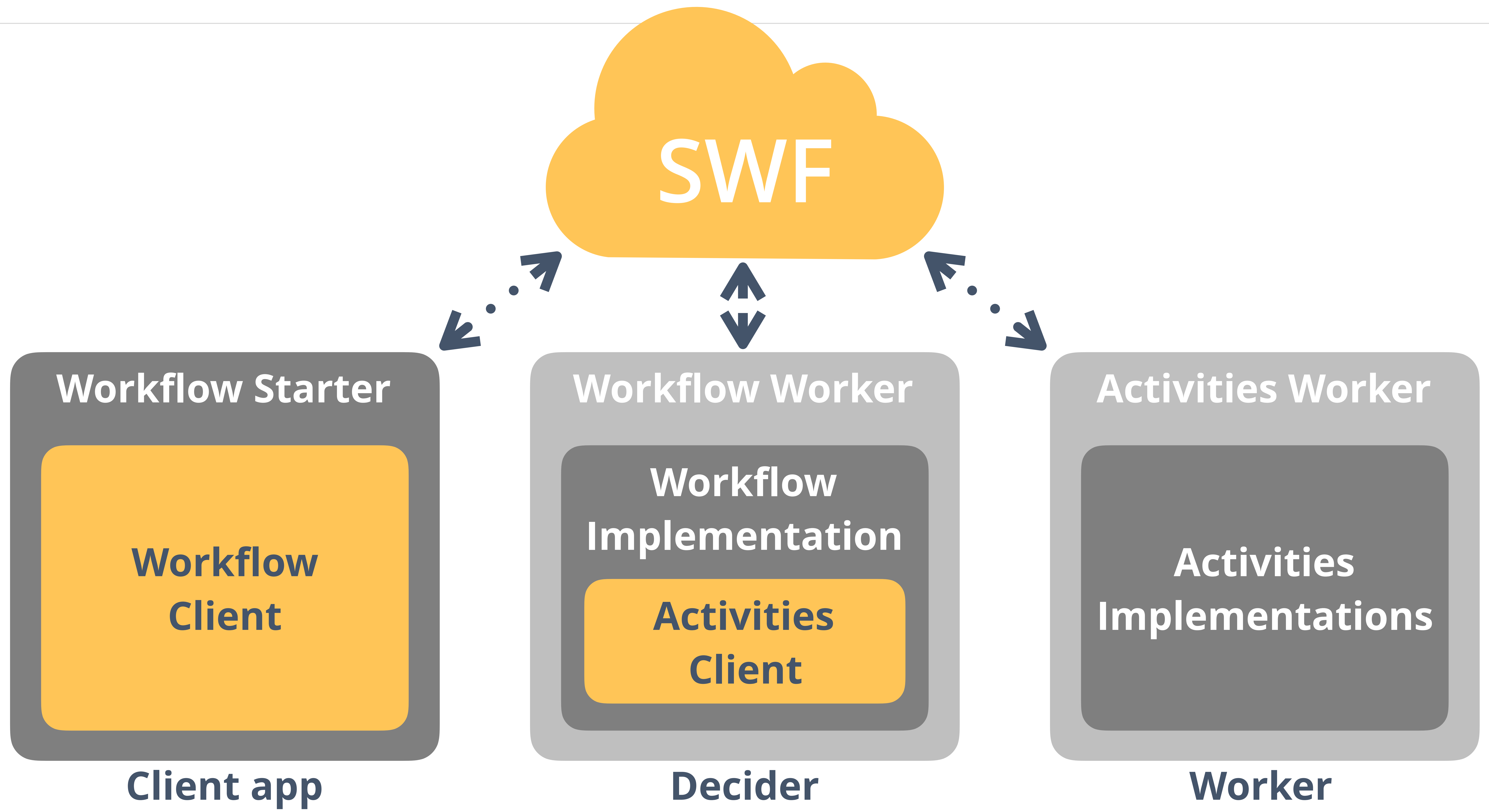
AWS SWF **Message Flow**



AWS SWF **Flow Framework**

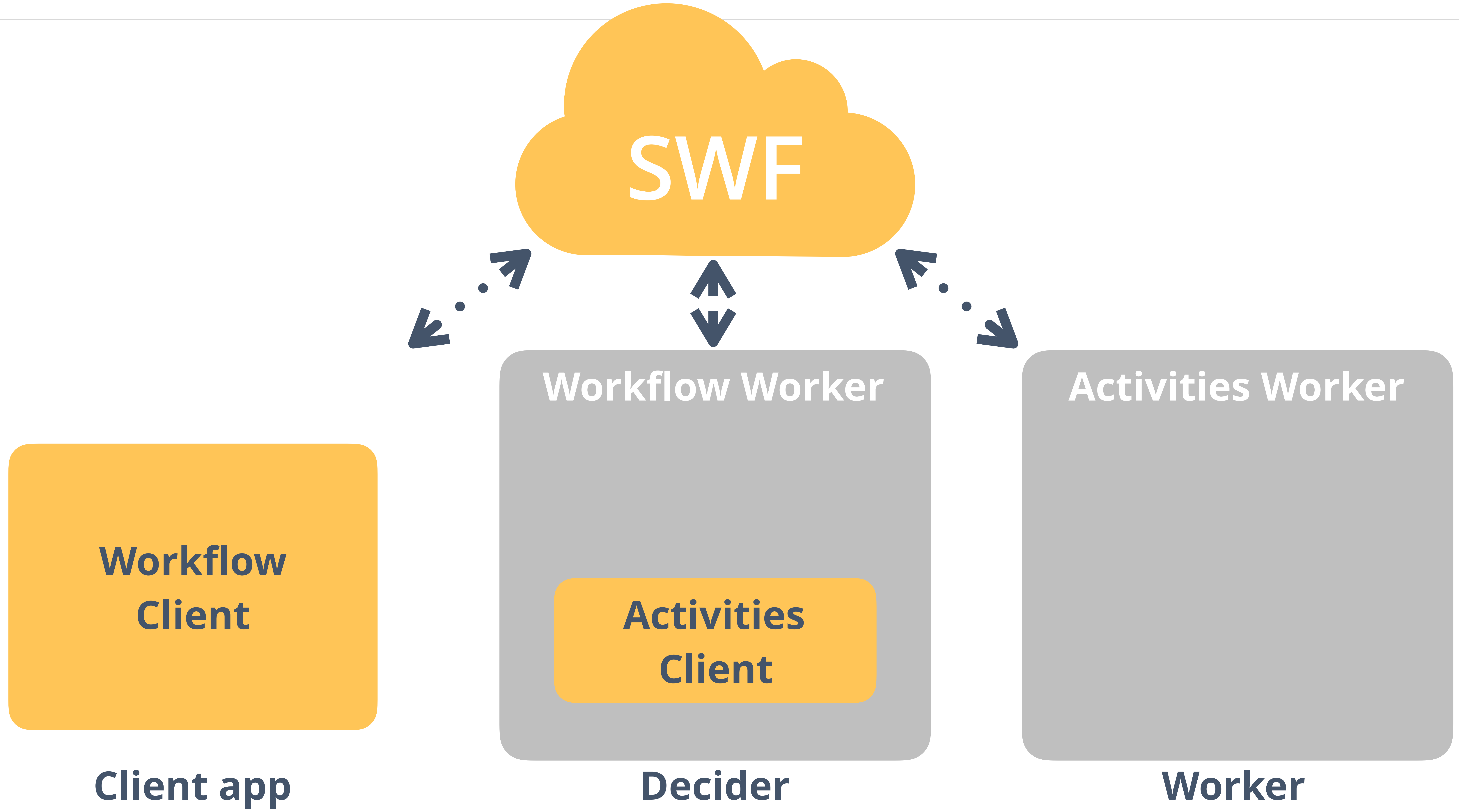


Java Flow Framework



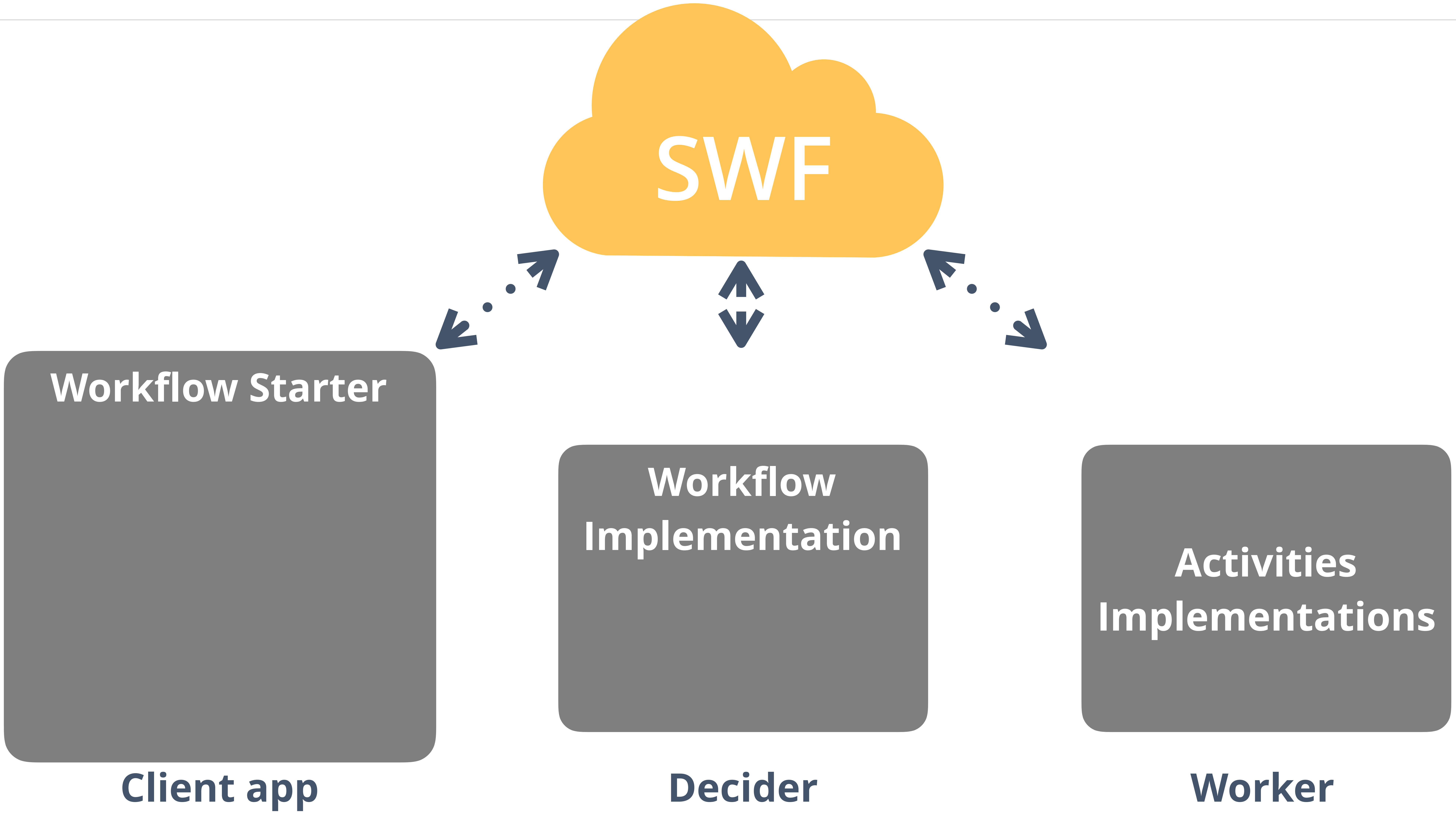
http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-application-structure.html

Java Flow Framework



http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-application-structure.html

Java Flow Framework



http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-application-structure.html

Java Flow Framework **Classes**

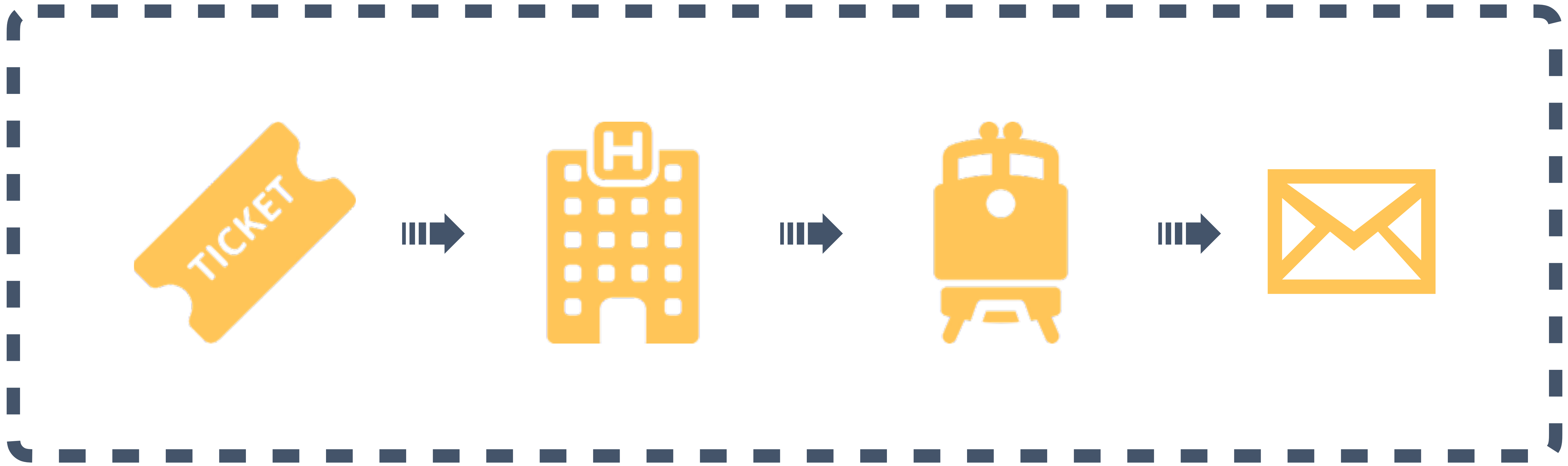
@Workflow	→	Client to schedule workflows
@Activities	→	Client to schedule activities

Java Flow Framework **Classes**

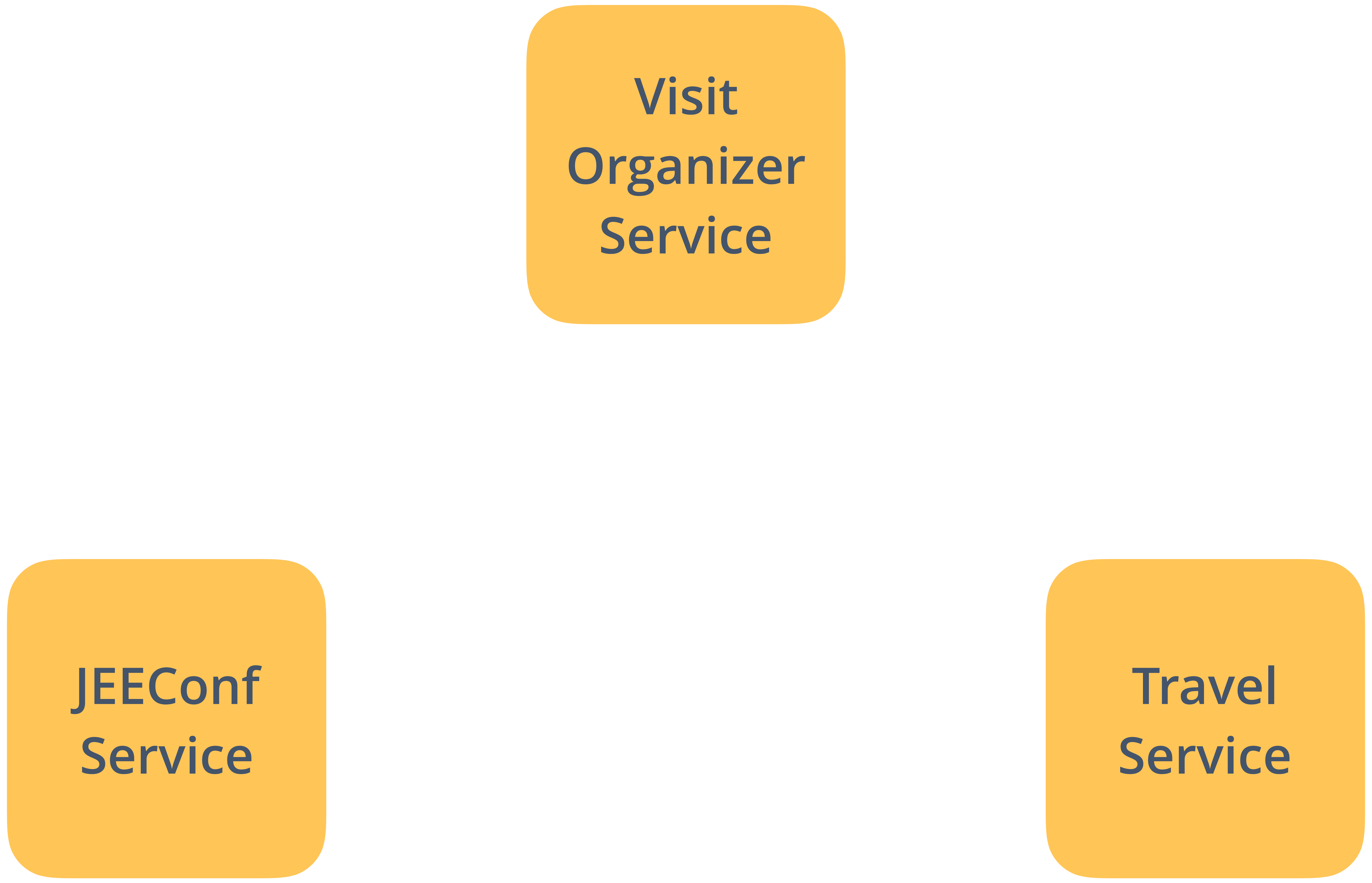
@Workflow	→	Client to schedule workflows
@Activities	→	Client to schedule activities
WorkflowWorker	→	Handles decision tasks
ActivitiesWorker	→	Handles activity tasks

SWF Demo

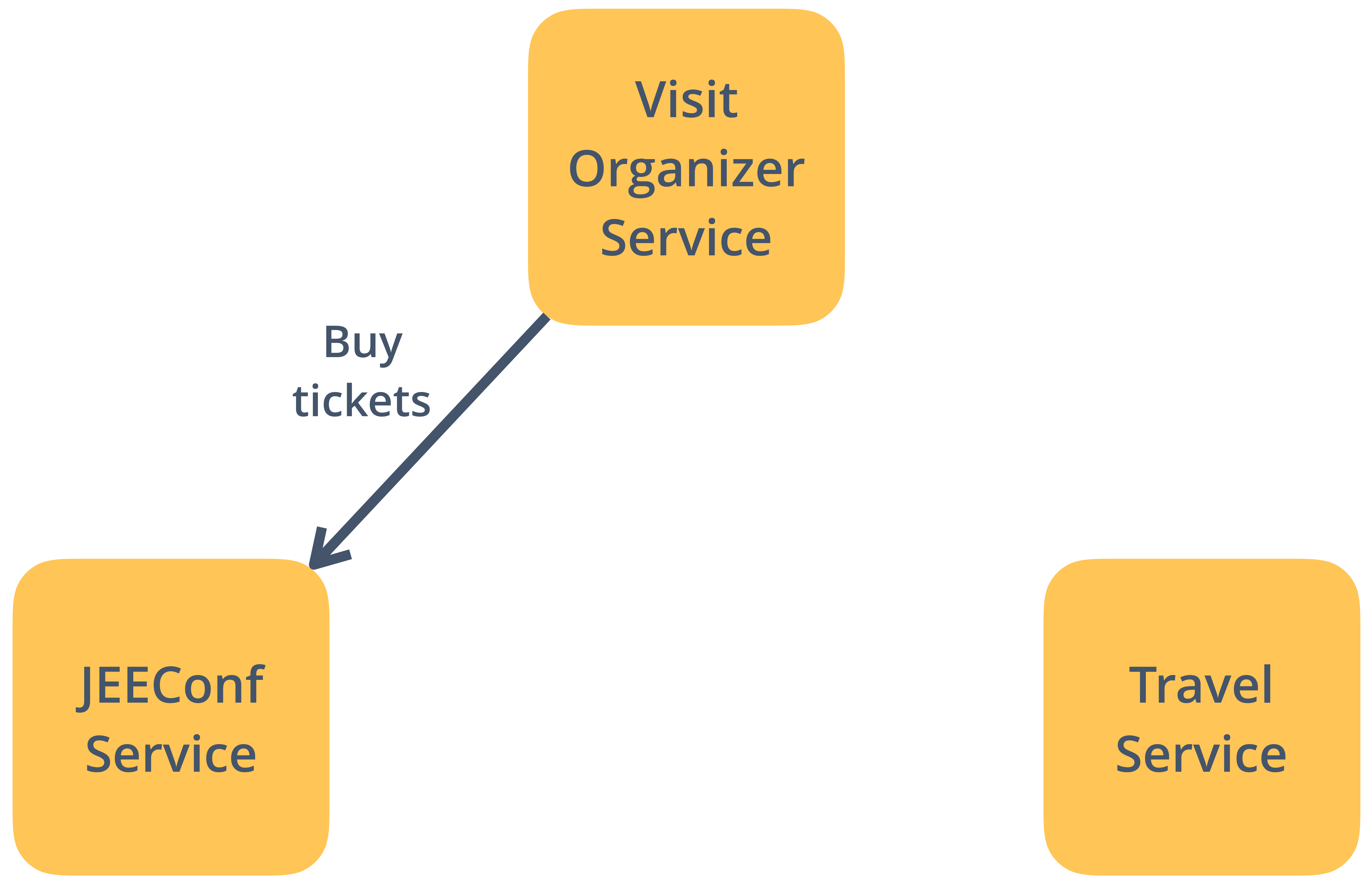
Visit Organizer **Workflow**



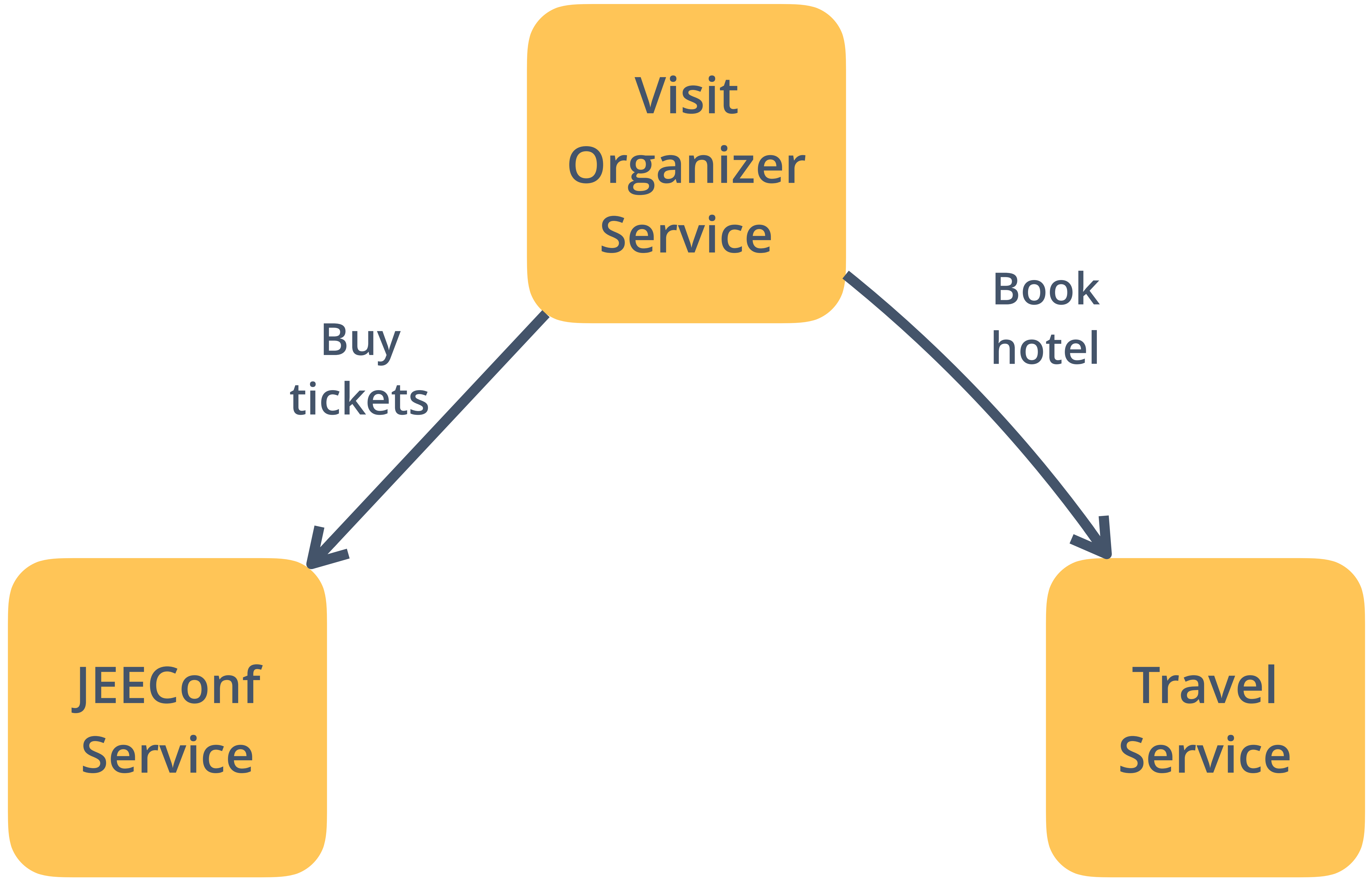
Preparing for **JEEConf**



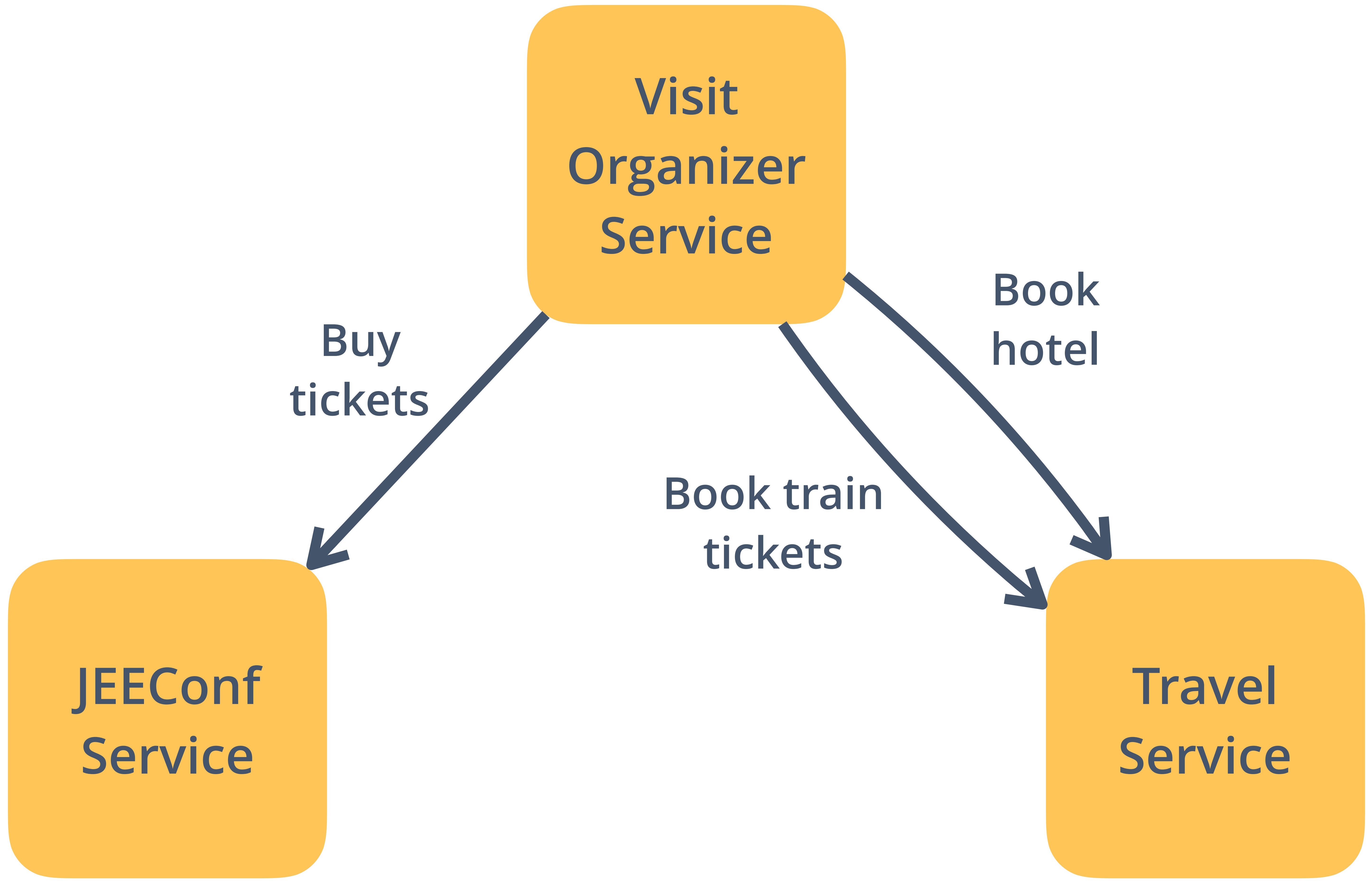
Preparing for **JEEConf**



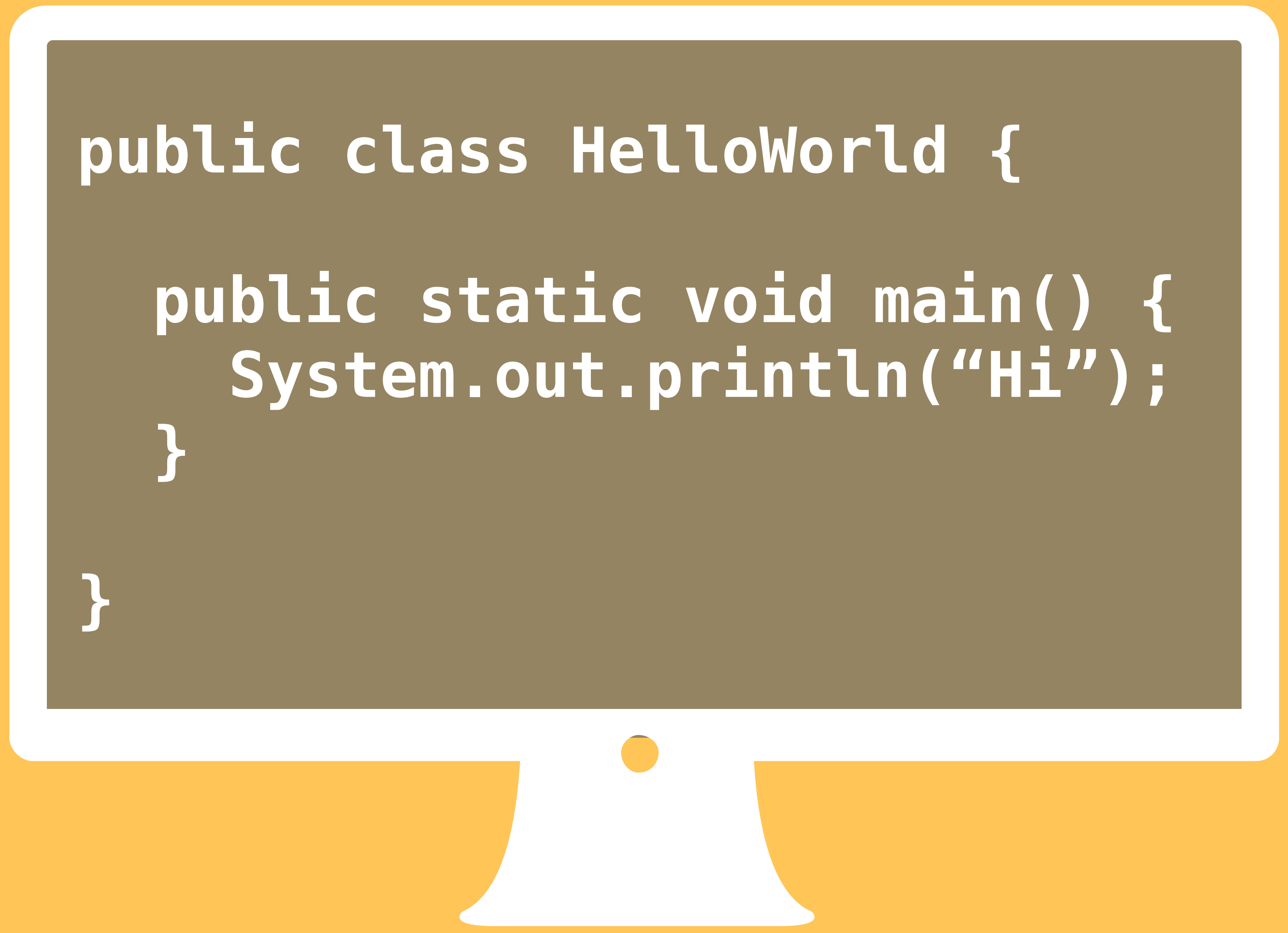
Preparing for **JEEConf**



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

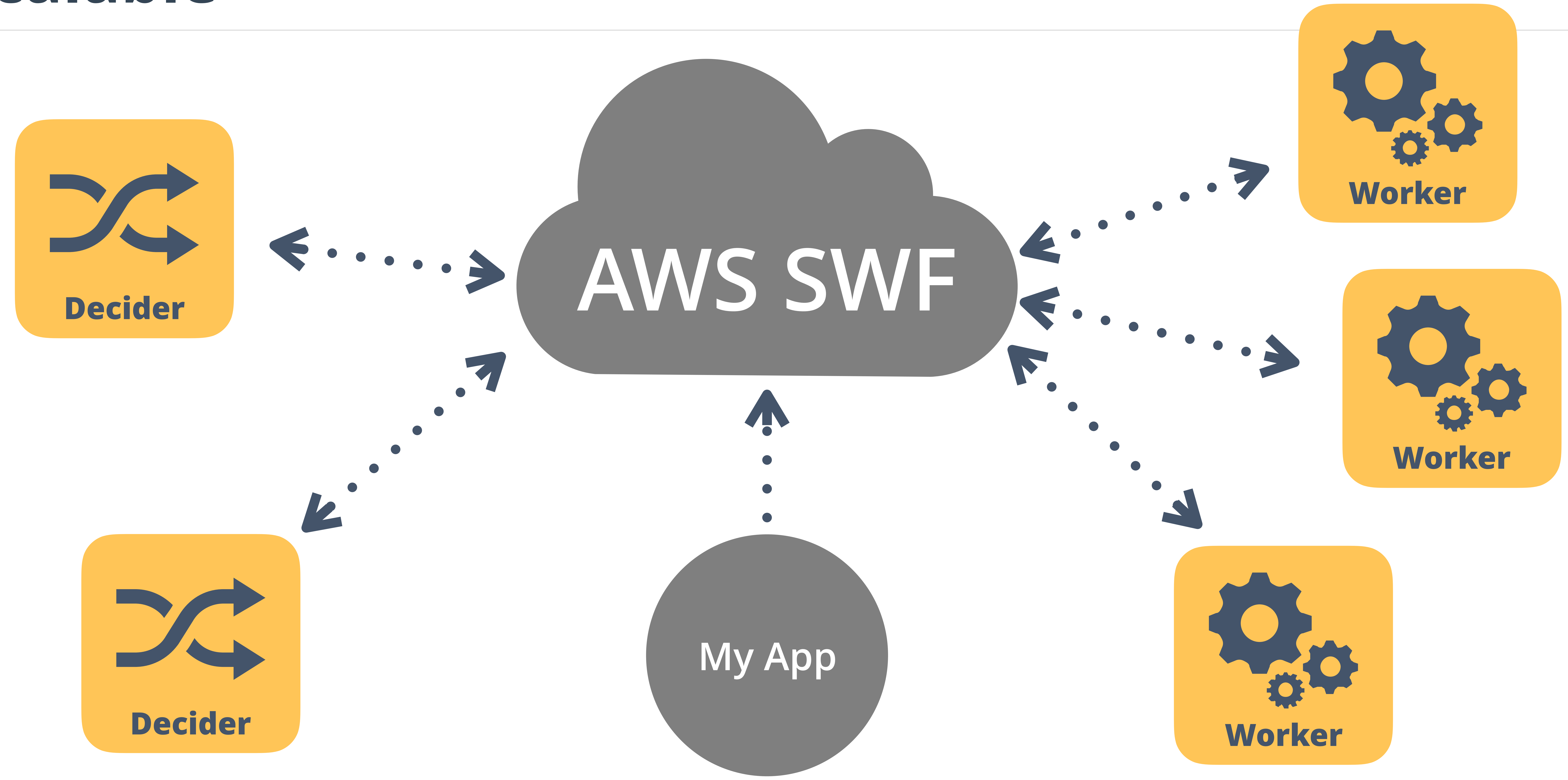
EASY, RIGHT?

http://info.thinkfun.com/stem-education/mark-engelberg-inventor-of-code-master-featured-in-getting-smart

SWF Features

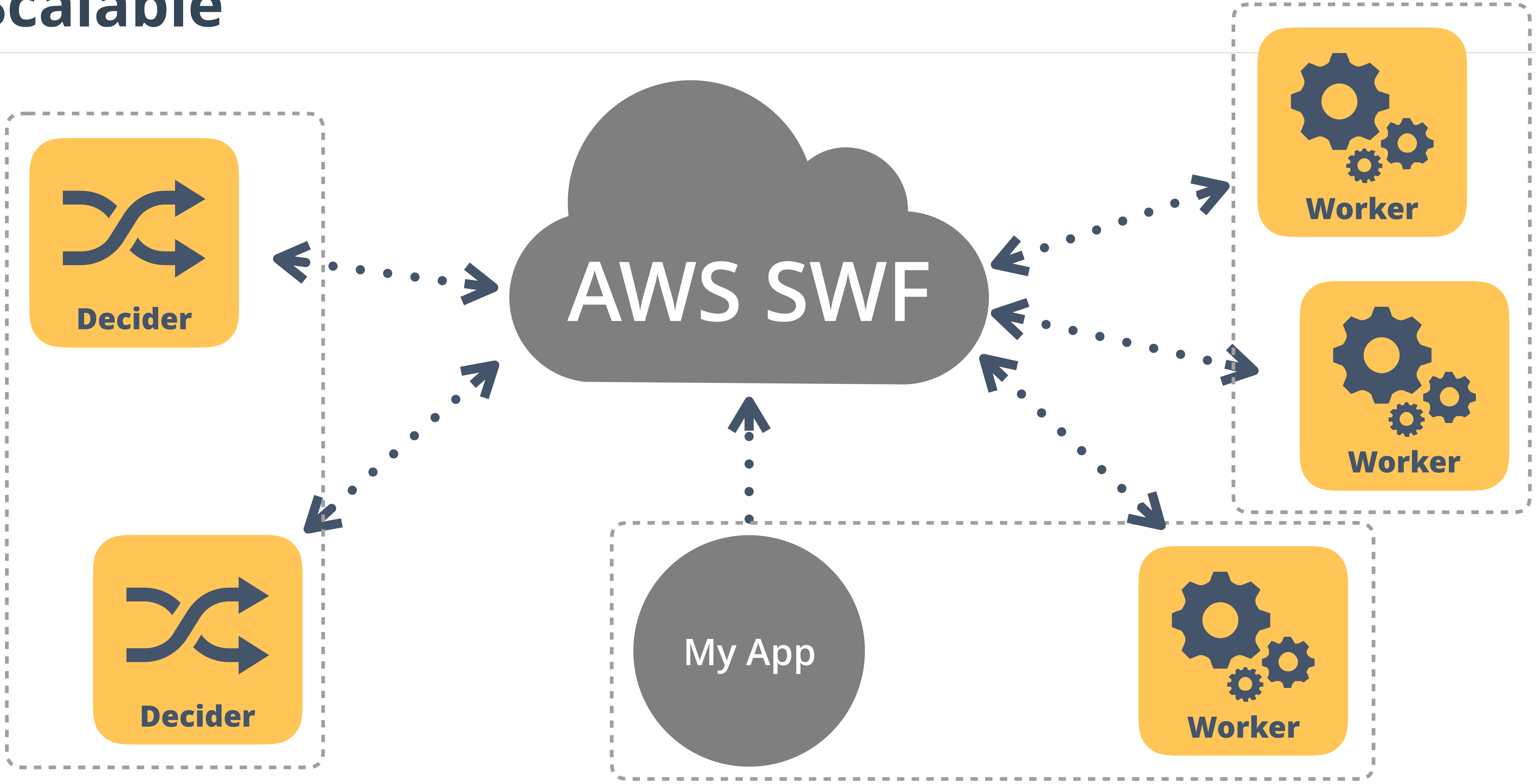
Fully Managed Service

Scalable



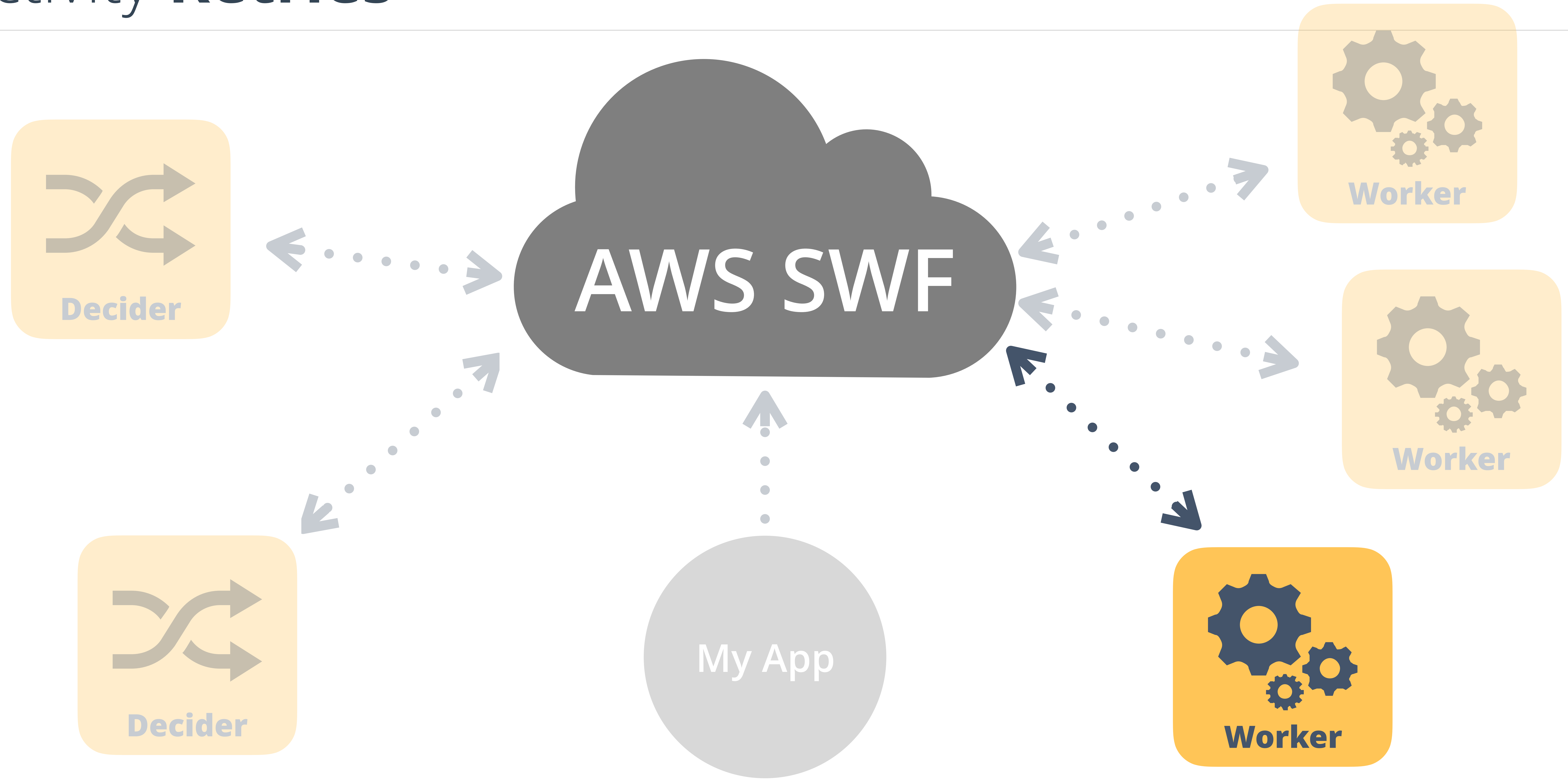
http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-scalable.html

Scalable



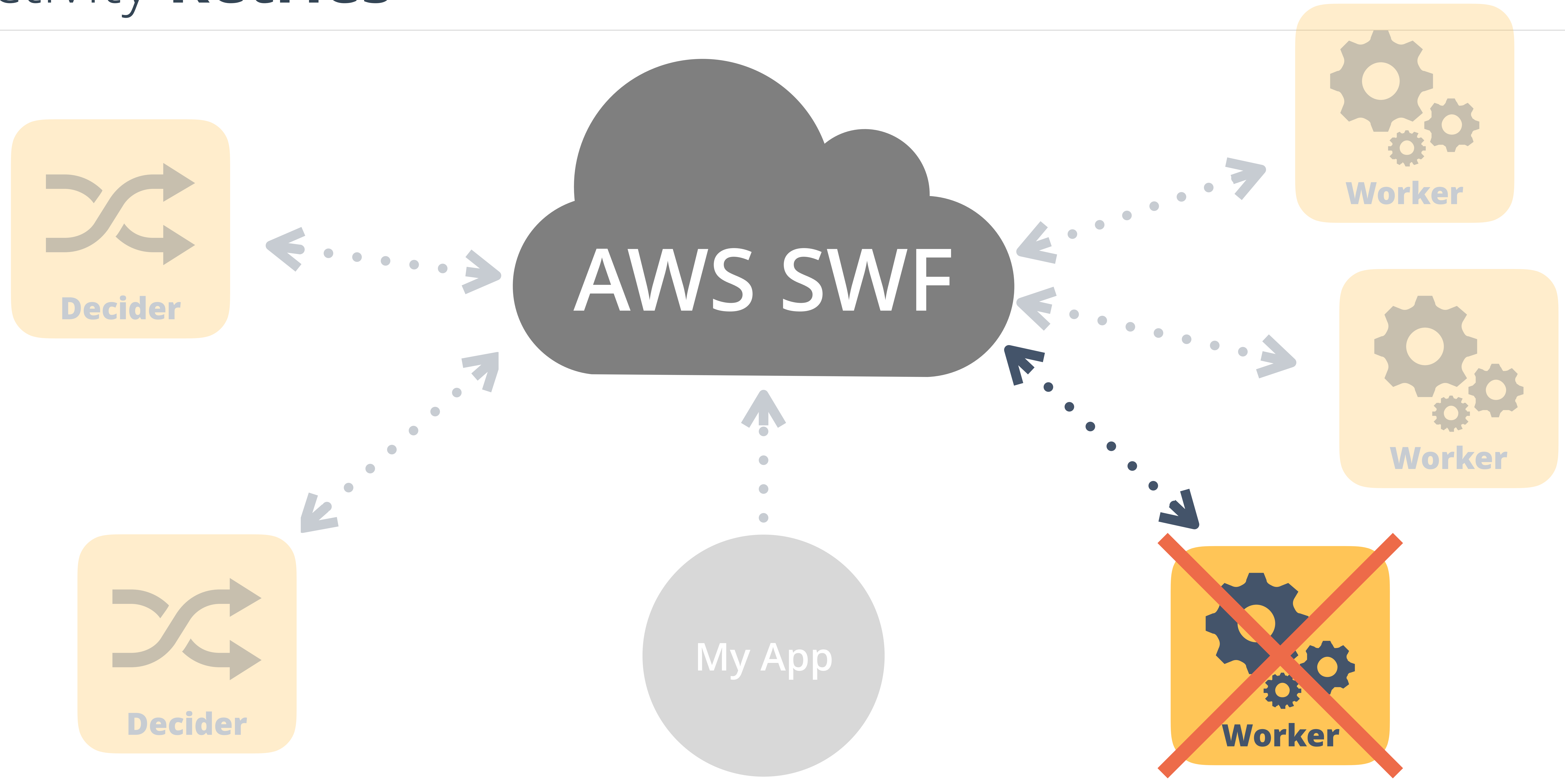
http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-scalable.html

Activity **Retries**



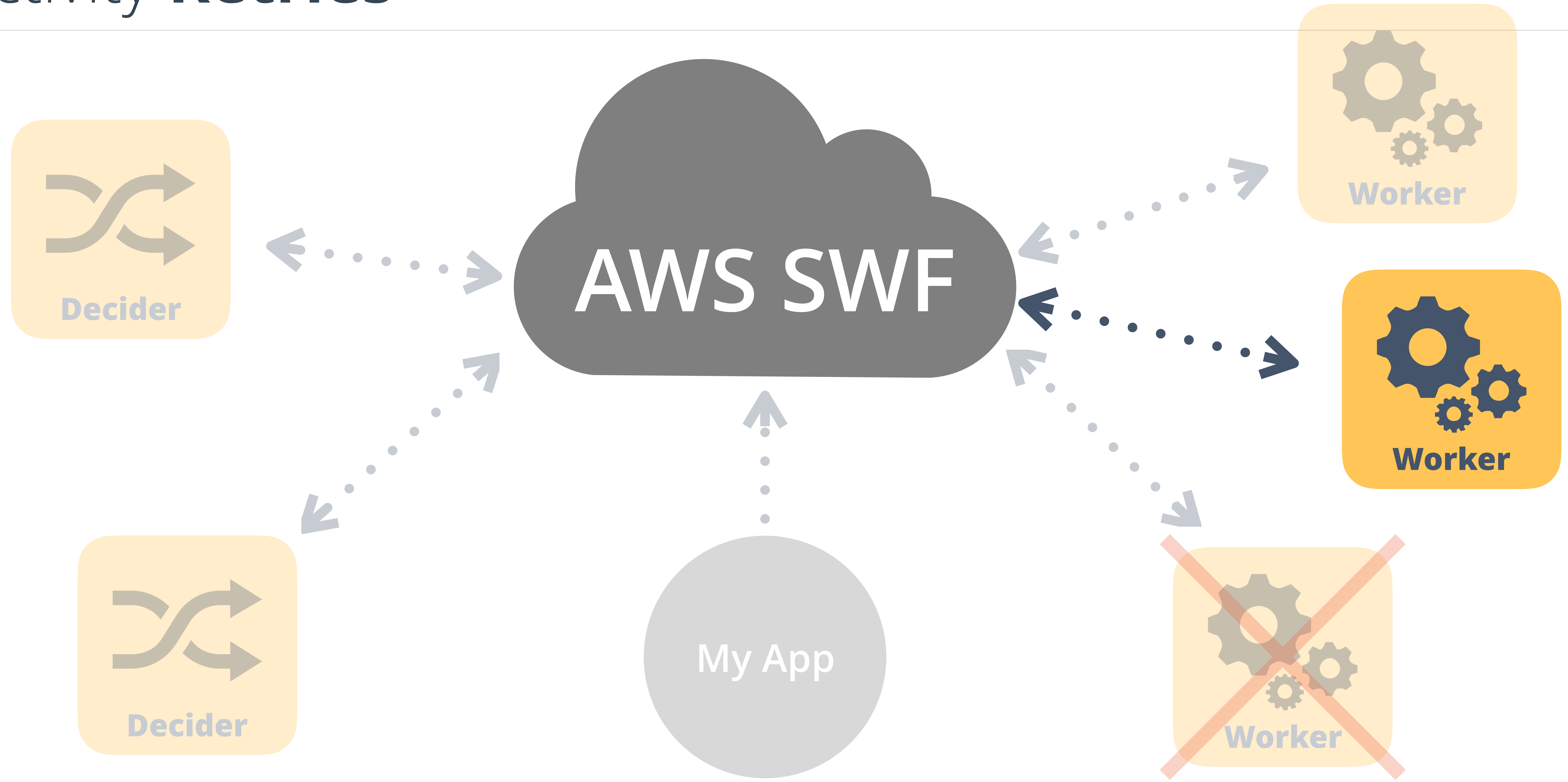
http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/features-retry.html

Activity **Retries**



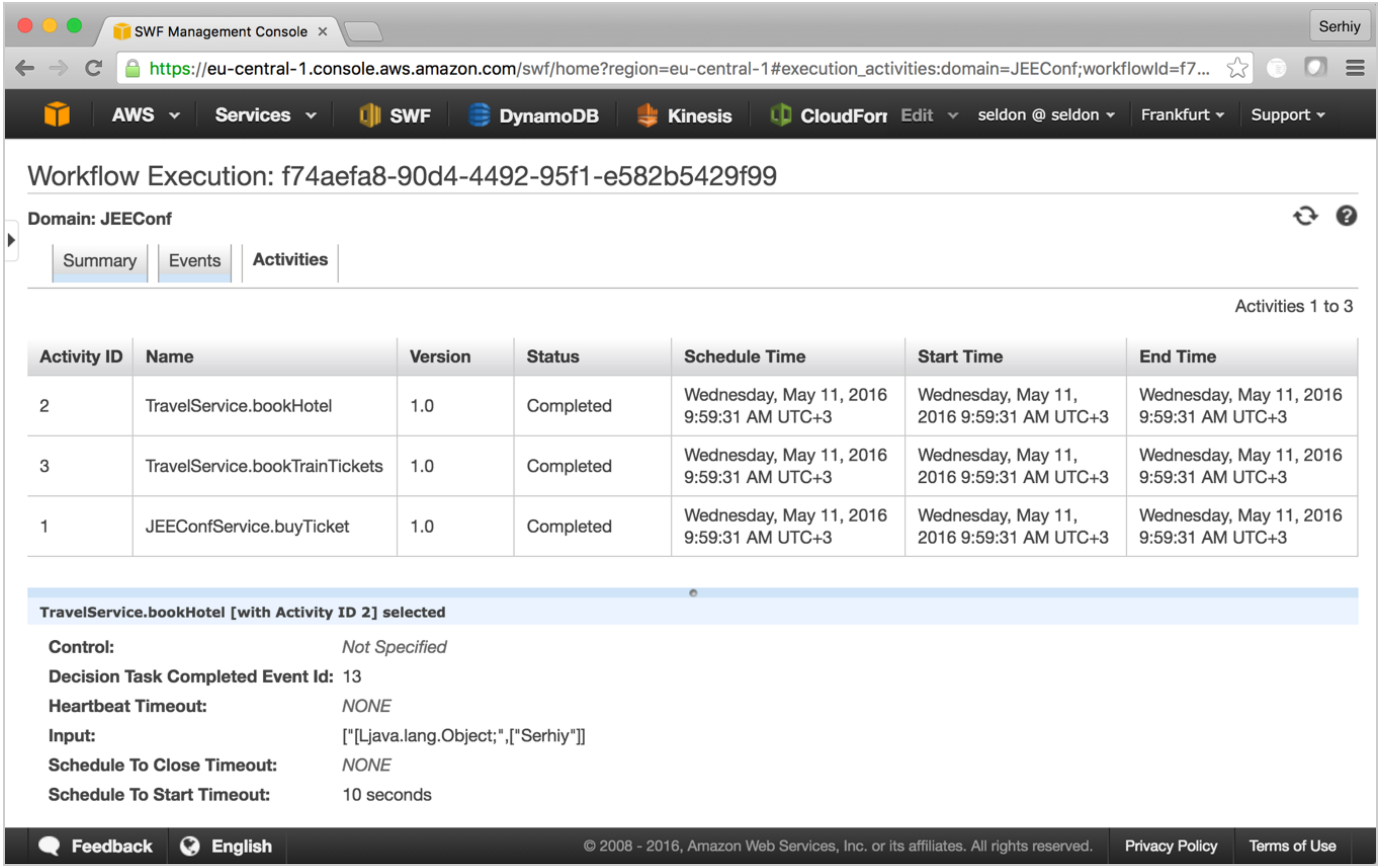
http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/features-retry.html

Activity **Retries**



http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/features-retry.html

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
CloudForı
Edit
seldon @ seldon
Frankfurt
Support
Summary
Events
Activities
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

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

Domain: JEEConf

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

Exactly Once Delivery

✉ = 1

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

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/test.html

AWS SWF **Pricing**

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

$1.75

https://aws.amazon.com/swf/pricing

SWF

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

Workflow Method **Replay #1**

Schedule

```java
@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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

Workflow Method **Replay #2**

Get Result

```java
@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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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);
}
```

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

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

http://docs.aws.amazon.com/amazonswf/latest/awsflowguide/awsflow-basics-distributed-execution.html

Activity **Timeouts**

Activity **Heartbeats**

Activity **Heartbeats**

History and Data Size **Limits**

25K

History events

32K

Message size (chars)

http://docs.aws.amazon.com/amazonswf/latest/developerguide/swf-dg-limits.html

Summary

Programs with steps

Scalability

Timeouts & Retries

Java & Ruby

Questions

Serhiy Batyuk
sbatyuk@gmail.com
@sbatyuk