



Pepper x Azure Cloud Robotics APIハンズオンワークショップ ～Microsoft Azureを簡単に使いこなそう～

アジェンダ

	翻訳API	会話理解API
1 デモ	●	●
2 Cloud Robotics APIの仕組み		
2 Cloud Robotics APIを利用したアプリ開発		
3 【Step.1】 日本語を英語に翻訳する	●	
4 【Step.2】 日本語でホテルを予約する		●
5 【Step.3】 英語でホテルを予約する	●	●

本日のハンズオンのスコープ

顔認識APIと写真説明APIをハンズオンの対象とします。

Cloud Robotics API の提供

Azure が無くても、Azure を知らなくても、Choregraphe さえあれば、容易に接続！
API 利用の為にコードは全て同じ手順、通信フォーマットが変わるだけ

翻訳 API

「この商品について、詳しく教えて欲しいのですが」



「想要知道更多關於這種產品。」

写真説明 API



「写真のためにポーズをとる人々のグループのようです」

「屋外に、5人、それぞれの性別と年齢は、.....です」



Device to Device API

「吉田様、ルーム #10をお使いください。今、部屋の鍵をお開きしました」



顔認識 API



「眼鏡お似合いですね。少々無精ひげも、お疲れですか？」

「サティアさん、あなたは、5回目の来日ですね」

会話理解 API (デモ版)



「ホテルを予約されたいのですね。それでは、予約に必要な事をお尋ねします」

「お探しのものは、2Fにあります。フロア地図を表示いたします」

メイン

時間があれば



デモ

サンプルアプリのデモ

① 翻訳

人が日本語で話しかけて、Pepperが英語で応える

人が英語で話しかけて、Pepperが日本語で応える

② 会話理解

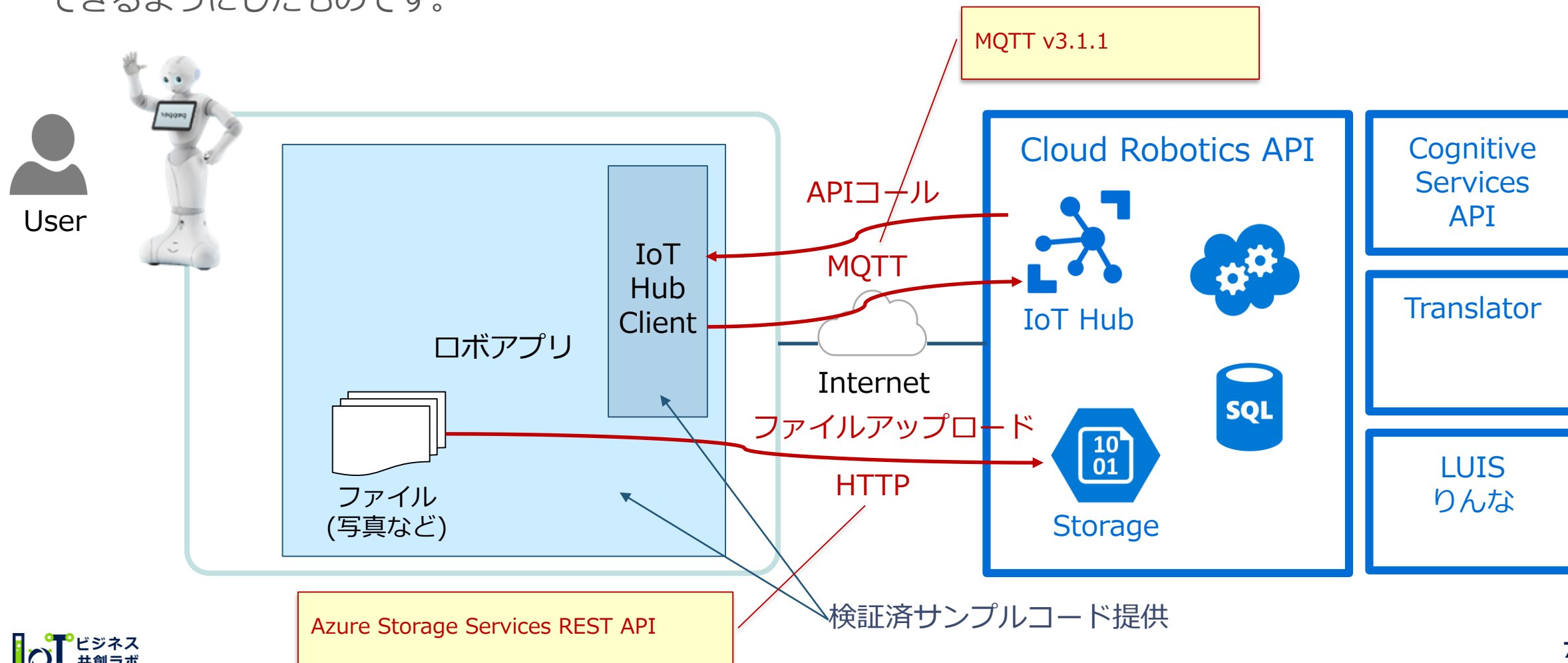
日本語でホテルを予約をする

英語でホテルを予約する

Cloud Robotics APIの仕組み

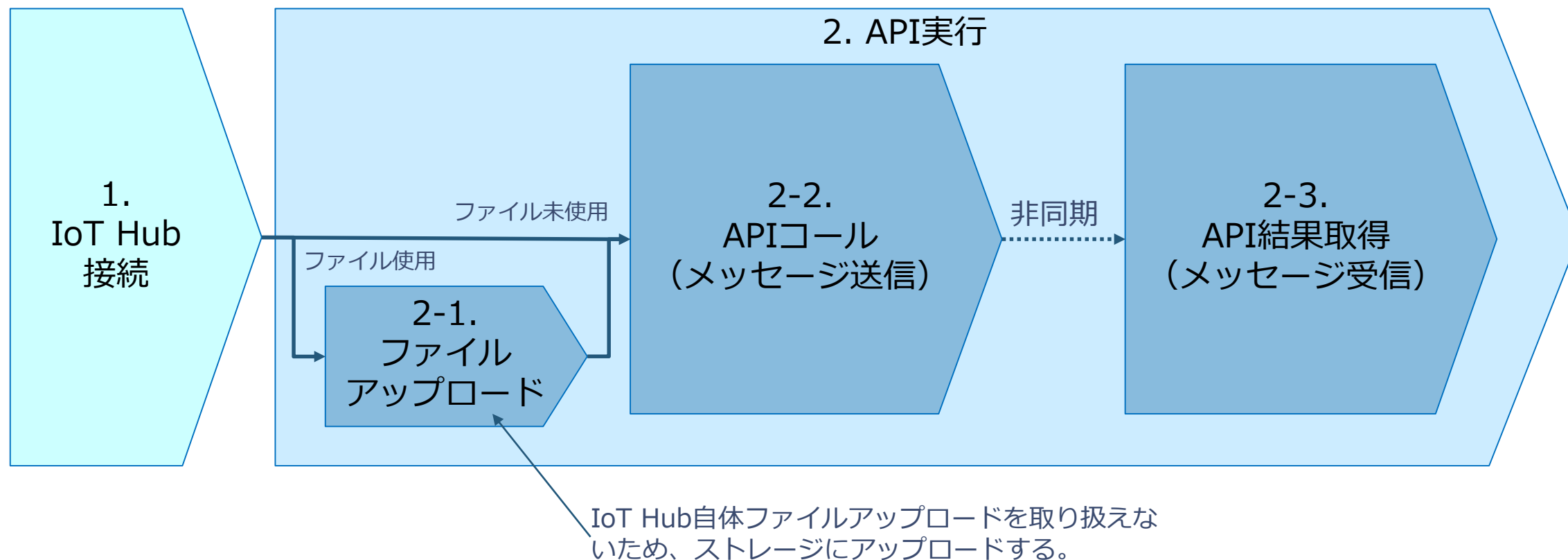
Cloud Robotics API

Cloud Robotics APIは、MS社提供の各種APIやサービスをラップし、共通の通信フォーマットで利用できるようにしたものです。



Cloud Robotics APIを使う時の処理フロー

APIを利用すると、以下のような処理フローになります。





Cloud Robotics APIを利用したアプリ開発

検証済コードの利用

Cloud Robotics APIを使った開発を簡易化するためのサンプルコードを利用することができます。

サンプルコード	使用目的	概要	Pythonモジュール・クラス・関数
API Client	APIをコールするため	IoT HubとMQTTで通信することができる。 APIのメッセージフォーマットに準じた処理が実装できる。	cloudrobotics.client CRFXClient cloudrobotics.message CRFXMessage
Blob Storage Uploader	写真などのファイルをAPIのインプットとして与えるため	Blob Storageにファイルをアップロードすることができる。	cloudrobotics.storage upload_to_storage()

Cloud Robotics APIとの接続

Cloud Robotics APIとの接続は以下のようなコードを書きます。

- API ClientとなるCRFXClientクラスを利用します。Azure IoT Hubのホスト名、デバイスID、デバイスのSharedAccessKeyが必要となります。
- CRFXClientのstart()メソッドで接続を開始します。接続に失敗した場合は5回リトライします。
- CRFXClientにコールバックを指定して、処理を実装します。

```
import cloudbotics.client as crfx
client = crfx.CRFXClient(' <Your Azure IoT Hub Hostname>', ' <Your Device Id>', ' <Your Device Key>')
# コールバックの設定
client.on_connect_successful = on_connect_successful
client.on_connect_failed = on_connect_failed
client.on_disconnect = on_disconnect
client.on_message = on_message
client.on_publish = on_publish
# 接続開始
client.start()
```

on_connect_successful	接続成功時
on_connect_failed	接続失敗時
on_disconnect	切断時
on_message	メッセージ受信時
on_publish	メッセージ送信時

Cloud Robotics APIへのメッセージ送信

メッセージ送信は以下のようなコードを書きます。

- CRFXClient、CRFXMessageクラスを利用します。
- CRFXMessageクラスはRbHeader、RbBodyの値をdictionaryで保持します。APIのメッセージ仕様に合わせて値をセットしてください。 ※Cloud Robotics APIの仕様書参照
- CRFXClientのsend_message()メソッドをコールし、メッセージを送信します。
- RoutingKeyword、AppId、MessageSeqno（送信連番）、SendDateTime（送信日時）は設定不要です。
- 戻りのメッセージとの突合せが必要な場合は、MessageSeqnoを使用してください。

メッセージオブジェクトの生成

```
message = message.CRFXMessage()  
message.header['RoutingType'] = 'CALL'  
message.header['AppProcessingId'] = 'RbAppFaceApi'  
message.header['MessageId'] = 'init'  
message.body['...'] = '....'
```

メッセージの送信（自動発行された送信連番が返却される）

```
seqNo = client.send_message(message)
```

Cloud Robotics APIからのメッセージ受信

メッセージ受信は以下のようなコードを書きます。

- CRFXClientクラスを利用します。
- CRFXClientクラスのon_messageにコールバックを指定します。

```
# API Clientにメッセージ受信時のコールバックをセット
client.on_message = on_message

# メッセージ受信時の処理
def on_message(received_message):
    if received_message.header['MessageId'] == 'init': # APIの種類=MessageIdにより分岐
        account = received_message.body['storageAccount']
        ...
```

Cloud Robotics APIと連携するためのBox

Choregrapheで作成するBoxでは、以下のようなコードを書きます。

```
def onInput_onStart(self): # Boxの開始でAPIクライアントの生成・コールバック設定・接続開始を実施
    import cloudrobotics.client as crfx
    self.client ← crfx.CRFXClient(<Your Azure IoT Hub's Hostname>, <Your Device Id>, <Your Device Key>)
    self.client.on_connect_successful = self.on_connect_successful
    self.client.on_message = self.on_message
    self.client.start()

def onInput_onCall(self): # Cloud Robotics APIをコールする
    message = CRFXMessage()
    ...
    self.client.send_message(message)

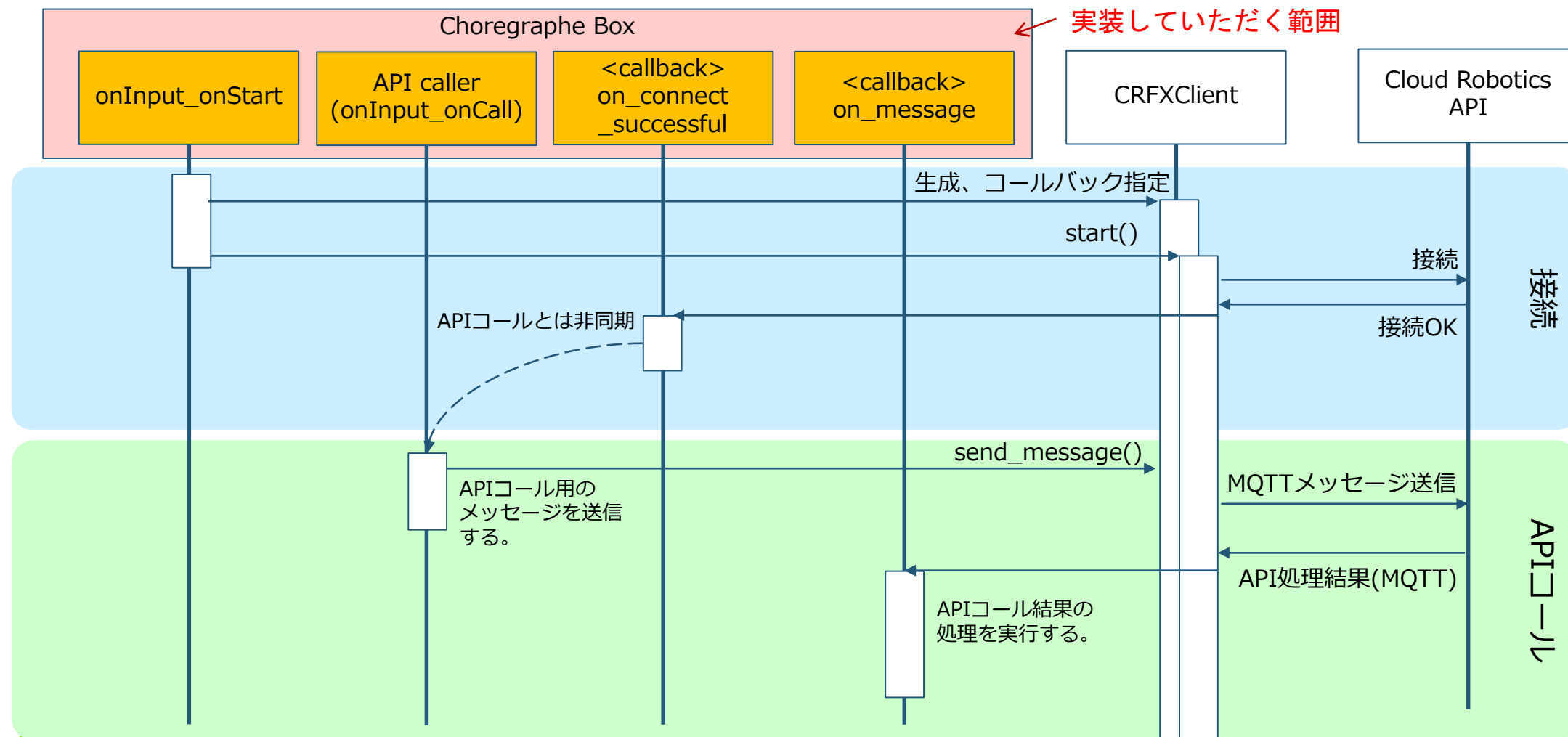
def on_connect_successful(self): # 接続成功時の処理
    ...

def on_message(self, received_message): # APIコール結果(メッセージ受信時)の処理
    ...
```

API ClientはBox(Class)のインスタンス変数としてください。
ローカル変数とすると、メッセージ受信ができません。
また、使用中はBoxは破棄されないように、
性質がonStoppedの出力を実行しないでください。

Cloud Robotics APIのコールシーケンス

Cloud Robotics APIをコールするためのシーケンス例は以下のとおり。



気を付けるポイント

1. メッセージングなので非同期処理です。
REST APIのようにAPIコール結果を同期的に取得できません。
2. りんなとの会話モードに入った場合は、「バイバイ」と話しかけて会話モードを終了させてください。

ハンズオン

ハンズオンを始めるにあたり

1. お手元に ChoregrapheがインストールされたPC をご用意ください。
2. Pepperを1台ずつご利用いただきます。
3. API仕様書、ハンズオン用サンプルコード をご準備ください。
4. APIの接続情報をご準備ください。
 1. Azure IoT Hubホスト名 `pephackiothub.azure-devices.net`
 2. デバイスID
 3. デバイスキー



【Step.1】日本語を英語に翻訳する

課題: 1

- hanson2_1配下のbehavior.xarを選択します。
- API仕様書の「通信フォーマット – 翻訳 API」のメッセージを送信し、結果のメッセージを受信して、日本語でPepperに話しかけてください。Pepperが英語で返答してきたら成功です。

```
8
9 def onLoad(self):
10     import cloudrobotics.client as crfx
11     import cloudrobotics.message as message
12
13     self.client = None
14     self.memory = ALProxy('ALMemory')
15     self.memory.insertData('CloudroboticsApiHandson02/Translator/FromLanguage', 'ja')
16     self.memory.insertData('CloudroboticsApiHandson02/Translator/ToLanguage', 'en')
17
18     # TODO: 【Step.1】 1.API Clientの生成
19     self.azure_iot_hub_hostname = '<...>'
20     self.azure_iot_hub_device_id = '<...>'
21     self.azure_iot_hub_device_key = '<...>'
22
```

インスタンス変数にIoT Hub接続情報を指定してください。

課題: 1

- ・ 翻訳APIおよび、PepperのDialog/Sayにそれぞれ設定する言語を指定します。複数ボックスにわたって利用するため、ALMemoryに値を設定します。

```
26
27     self.memory = ALProxy('ALMemory')
28     # TODO: 【Step.1】 2.翻訳する言語の設定
29     self.memory.insertData('CloudroboticsApiHandson02/Translator/FromLanguage', '<...>')
30     self.memory.insertData('CloudroboticsApiHandson02/Translator/ToLanguage', '<...>')
31
```

- ・ FromLanguageにjaという文字列を設定してください。
- ・ ToLanguageにenという文字列を設定してください。

課題: 1

- ・ API仕様書の「通信フォーマット -- 翻訳 API」の「デバイスからのクラウドへの送信時：D2C」に従って、RoutingTypeとAppProcessingIdを設定してください。

```
44 # TODO: 【Step.1】 3.日本語から英語へ翻訳
45 translator_message = message.CRFXMessage()
46 translator_message.header['RoutingType'] = '<...>'
47 translator_message.header['AppProcessingId'] = '<...>'
48
49 translator_message.body['visitor'] = '<...>'
50 translator_message.body['visitor_id'] = '<...>'
51 translator_message.body['text'] = '<...>'
52 translator_message.body['tolang'] = '<...>'
```

API仕様書の送信メッセージの定義に従い、値をセットしてください。

課題: 1

- ・ 翻訳APIのRbBodyを設定します。

```
44 # TODO: 【Step.1】 3.日本語から英語へ翻訳
45 translator_message = message.CRFXMessage()
46 translator_message.header['RoutingType'] = '<...>'
47 translator_message.header['AppProcessingId'] = '<...>'
48
49 translator_message.body['visitor'] = '<...>'
50 translator_message.body['visitor_id'] = '<...>'
51 translator_message.body['text'] = '<...>'
52 translator_message.body['tolang'] = '<...>'
```

・ visitor/visitor_idには任意の値を設定してください。

・ textには、onInput_onCallTranslator関数の仮引数textを設定してください。

・ tolangには、ALMemoryの以下のキーから値を設定してください。
CloudroboticsApiHandson02/Translator/ToLanguage

課題: 1

- ・ 翻訳APIから受信したメッセージを処理します。

```
88
89 try:
90     # TODO: 【Step.1】 4.CloudroboticsApiの受信メッセージから翻訳APIのもののみを取得する
91     # メッセージヘッダーのAppId/AppProcessingIdに応じて処理を実装
92     if received_message.header['AppId'] == '<...>' and received_message.header['AppProcessingId'] == '<...>':
93         if received_message.body['success'] == 'true':
94             # TODO: 【Step.1】 5.翻訳されたメッセージを取得する
95             self.onSuccess('<...>')
96         else:
```

- ・ 処理するメッセージが翻訳APIからのものであることを判別するために、AppIdとAppProcessingIdに適切な値を設定してください。
- ・ 翻訳APIから受信したメッセージの<翻訳済みのテキスト>をonSuccessの引数に設定してください。

【Step.2】 日本語でホテルを予約する

課題: 2

- ・ hanson2_2配下のbehavior.xarを選択します。
- ・ API仕様書の「通信フォーマット -- 会話理解 API」のメッセージを送信し、結果のメッセージを受信して、日本語でPepperに「ホテルを予約したい」と話しかけてください。Pepperが宿泊日を確認してきたら成功です。

```
8
9 ☐ def onLoad(self):
10     import cloudrobotics.client as crfx
11     import cloudrobotics.message as message
12
13     self.client = None
14
15     # TODO: 【Step.2】 1.API Clientの生成
16     self.azure_iot_hub_hostname = '<...>'
17     self.azure_iot_hub_device_id = '<...>'
18     self.azure_iot_hub_device_key = '<...>'
19
```

} インスタンス変数にIoT Hub接続情報を指定してください。

課題: 2

- API仕様書の「通信フォーマット -- 会話理解 API - “init” (会話セッションの開始)」の「デバイスからのクラウドへの送信時：D2C」に従って、RoutingTypeとAppProcessingIdを設定してください。

```
36
37 ☐ def onInput_onCall(self, text):
38     import cloudrobotics.message as message
39
40     # TODO: 【Step.2】 2.会話セッションの開始/継続
41     conversation_message = message.CRFXMessage()
42     conversation_message.header['RoutingType'] = '<...>'
43     conversation_message.header['MessageId'] = '<...>'
44     conversation_message.header['AppProcessingId'] = '<...>'
45
46     conversation_message.body['visitor'] = '<...>'
47     conversation_message.body['visitor_id'] = '<...>'
48     conversation_message.body['talkByMe'] = '<...>'
49
```

API仕様書の送信メッセージの定義に従い、
値をセットしてください。

課題: 2

- ・ 会話理解 APIのRbBodyを設定します。

```
36
37 ☐ def onInput_onCall(self, text):
38     import cloudrobotics.message as message
39
40     # TODO: 【Step.2】 2.会話セッションの開始/継続
41     conversation_message = message.CRFXMessage()
42     conversation_message.header['RoutingType'] = '<...>'
43     conversation_message.header['MessageId'] = '<...>'
44     conversation_message.header['AppProcessingId'] = '<...>'
45
46     conversation_message.body['visitor'] = '<...>'
47     conversation_message.body['visitor_id'] = '<...>'
48     conversation_message.body['talkByMe'] = '<...>'
49
```

・ visitor/visitor_idには任意の値を設定してください。

・ talkByMeには、onInput_onCall関数の仮引数textを設定してください。

課題: 2

- ・ 会話理解APIから受信したメッセージを処理します。

```
85 try:
86     # TODO: 【Step.2】 3.CloudroboticsApiの受信メッセージから会話理解APIのもののみを取得する
87     # メッセージヘッダーのAppId/AppProcessingIdに応じて処理を実装
88     if received_message.header['AppId'] == '<...>' and received_message.header['AppProcessingId'] == '<...>':
89         if received_message.body['success'] == 'true' or received_message.body['type'] == 'finishTalk':
90             if received_message.body['type'] == 'init' or len(received_message.body['type']) >= 4 and received_m
91                 # TODO: 【Step.2】 4.AIからの応答メッセージの取得
92                 visitor = '<...>'
93                 visitor_id = '<...>'
94                 self.type = '<...>'
95                 talkByAi = '<...>'
96                 for talk in talkByAi:
97                     self.onSuccess('<...>')
```

- ・ 処理するメッセージが会話理解APIからのものであることを判別するために、AppIdとAppProcessingIdに適切な値を設定してください。

課題: 2

- ・ 会話理解APIから受信したメッセージを処理します。

```
85 try:
86     # TODO: 【Step.2】 3.CloudroboticsApiの受信メッセージから会話理解APIのもののみを取得する
87     # メッセージヘッダーのAppId/AppProcessingIdに応じて処理を実装
88     if received_message.header['AppId'] == '<...>' and received_message.header['AppProcessingId'] == '<...>':
89         if received_message.body['success'] == 'true' or received_message.body['type'] == 'finishTalk':
90             if received_message.body['type'] == 'init' or len(received_message.body['type']) >= 4 and received_m
91                 # TODO: 【Step.2】 4 AIからの応答メッセージの取得
92                 visitor = '<...>'
93                 visitor_id = '<...>'
94                 self.type = '<...>'
95                 talkByAi = '<...>'
96                 for talk in talkByAi:
97                     self.onSuccess('<...>')
```

・ visitor/visitor_idには任意の値を設定してください。

・ self.type/talkByAiには、API仕様書を参考に会話理解APIの受信メッセージから適切な値を設定してください。

・ onSuccessの引数に、会話理解APIの受信メッセージのtalkByAiからSayTextの値を設定してください。

課題: 2

- ・ 会話理解APIから受信したメッセージを処理します。

```
99 elif received_message.body['type'] == 'finishTalk':  
100     # TODO: 【Step.2】 5.AIからの終了メッセージ取得後の処理  
101     self.type = received_message.body['type'] = 'init'  
102     talkByAi = received_message.body['talkByAi']  
103     for talk in talkByAi:  
104         self.onSuccess('<...>')  
105
```

- ・ onSuccessの引数に、会話理解APIの受信メッセージのtalkByAiからSayTextの値を設定してください。

課題: 2

- ・ 会話理解APIから受信したメッセージを処理します。

```
106 elif received_message.body['type'] == 'launchDanceApp1':  
107     # TODO: 【Step.2】 6.AIからのダンスアプリ起動メッセージ取得後の処理  
108     self.onSuccess('<...>')  
109
```

- ・ onSuccessの引数に、' それでは、ダンスアプリを起動します' という文字列を設定してください。

【Step.3】 英語でホテルを予約する

課題: 3

- hanson2_3配下のbehavior.xarを選択します。
- 翻訳APIと会話理解APIを組み合わせ、人が英語で話しかけた内容を翻訳APIで日本語に翻訳し、会話理解APIに受け渡すアプリケーションを作成します。

```
12
13     self.client = None
14     self.memory = ALProxy('ALMemory')
15     self.memory.insertData('CloudroboticsApiHanson02/Translator/FromLanguage', 'en')
16     self.memory.insertData('CloudroboticsApiHanson02/Translator/ToLanguage', 'ja')
17
18     # TODO: 【Step.3】 1.API Clientの生成
19     self.azure_iot_hub_hostname = '<...>'
20     self.azure_iot_hub_device_id = '<...>'
21     self.azure_iot_hub_device_key = '<...>' }
```

インスタンス変数にIoT Hub接続情報を指定してください。

課題: 3

- ・ 翻訳APIおよび、PepperのDialog/Sayにそれぞれ設定する言語を指定します。複数ボックスにわたって利用するため、ALMemoryに値を設定します。

```
26
27     self.memory = ALProxy('ALMemory')
28     # TODO: 【Step.3】 2.翻訳する言語の設定
29     self.memory.insertData('CloudroboticsApiHandson02/Translator/FromLanguage', '<...>')
30     self.memory.insertData('CloudroboticsApiHandson02/Translator/ToLanguage', '<...>')
```

- ・ FromLanguageにenという文字列を設定してください。
- ・ ToLanguageにjaという文字列を設定してください。

課題: 3

・API仕様書の「通信フォーマット -- 翻訳 API」の「デバイスからのクラウドへの送信時：D2C」に従って、RoutingTypeとAppProcessingIdを設定してください。

```
43 def onInput_onCallTranslator(self, text):
44     import clouddrobotics.message as message
45
46     # TODO: 【Step.3】 3.英語から日本語へ翻訳
47     translator_message = message.CRFXMessage()
48     translator_message.header['RoutingType'] = '<...>'
49     translator_message.header['AppProcessingId'] = '<...>'
50
51     translator_message.body['visitor'] = '<...>'
52     translator_message.body['visitor_id'] = '<...>'
53     translator_message.body['text'] = '<...>'
54     translator_message.body['tolang'] = '<...>'
```

API仕様書の送信メッセージの定義に従い、値を設定してください。

課題: 3

- ・ 翻訳APIのRbBodyを設定します。

```
43 def onInput_onCallTranslator(self, text):
44     import cloudrobotics.message as message
45
46     # TODO: 【Step.3】 3.英語から日本語へ翻訳
47     translator_message = message.CRFXMessage()
48     translator_message.header['RoutingType'] = '<...>'
49     translator_message.header['AppProcessingId'] = '<...>'
50
51     translator_message.body['visitor'] = '<...>'
52     translator_message.body['visitor_id'] = '<...>'
53     translator_message.body['text'] = '<...>'
54     translator_message.body['tolang'] = '<...>'
```

・ visitor/visitor_idには任意の値を設定してください。

・ textには、onInput_onCallTranslator関数の仮引数textを設定してください。

・ tolangには、ALMemoryの以下のキーから値を設定してください。
CloudroboticsApiHandson02/Translator/ToLanguage

課題: 3

・API仕様書の「通信フォーマット -- 会話理解 API - “init” (会話セッションの開始)」の「デバイスからのクラウドへの送信時：D2C」に従って、RoutingTypeとAppProcessingIdを設定してください。

・API仕様書の送信メッセージの定義に従い、値をセットしてください。

・MessageIdには、インスタンス変数self.typeをセットしてください。

```
68 # TODO: 【Step.3】 4.会話セッションの開始/継続
69 conversation_message = message.CRFXMessage()
70 conversation_message.header['RoutingType'] = '<...>'
71 conversation_message.header['MessageId'] = '<...>'
72 conversation_message.header['AppProcessingId'] = '<...>'
73
74 conversation_message.body['visitor'] = '<...>'
75 conversation_message.body['visitor_id'] = '<...>'
76 conversation_message.body['talkByMe'] = '<...>'
```

課題: 3

- ・ 会話理解 APIのRbBodyを設定します。

```
68 # TODO: 【Step.3】 4.会話セッションの開始/継続
69 conversation_message = message.CRFXMessage()
70 conversation_message.header['RoutingType'] = '<...>'
71 conversation_message.header['MessageId'] = '<...>'
72 conversation_message.header['AppProcessingId'] = '<...>'
73
74 conversation_message.body['visitor'] = '<...>'
75 conversation_message.body['visitor_id'] = '<...>'
76 conversation_message.body['talkByMe'] = '<...>'
```

・ visitor/visitor_idには任意の値を設定してください。

・ talkByMeには、callConversation関数の仮引数textを設定してください。

課題: 3

- ・ 翻訳APIから受信したメッセージを処理します。

```
108 try:
109     # TODO: 【Step.3】 5.CloudroboticsApiの受信メッセージから翻訳APIのもののみを取得する
110     # メッセージヘッダーのAppId/AppProcessingIdに応じて処理を実装
111     if received_message.header['AppId'] == '<...>' and received_message.header['AppProcessingId'] == '<...>':
112         if received_message.body['success'] == 'true':
113             # TODO: 【Step.3】 6.翻訳されたメッセージを会話理解APIへ送信する
114             self.callConversation('<...>')
115         else:
```

- ・ 処理するメッセージが翻訳APIからのものであることを判別するために、AppIdとAppProcessingIdに適切な値を設定してください。
- ・ 翻訳APIから受信したメッセージの<翻訳済みのテキスト>をcallConversationの引数に設定してください。

課題: 3

- ・ 会話理解APIから受信したメッセージを処理します。

```
118 # TODO: 【Step.3】 7.CloudroboticsApiの受信メッセージから会話理解APIのもののみを取得する
119 elif received_message.header['AppId'] == '<...>' and received_message.header['AppProcessingId'] == '<...>':
120     if received_message.body['success'] == 'true' or received_message.body['type'] == 'finishTalk':
121         if received_message.body['type'] == 'init' or len(received_message.body['type']) >= 4 and received_message
122             # TODO: 【Step.3】 8.AIからの応答メッセージの取得
123             visitor = '<...>'
124             visitor_id = '<...>'
125             self.type = '<...>'
126             talkByAi = '<...>'
127             for talk in talkByAi:
128                 self.onSuccess('<...>')
```

・ 処理するメッセージが会話理解APIからのものであることを判別するために、AppIdとAppProcessingIdに適切な値を設定してください。

課題: 3

- ・ 会話理解APIから受信したメッセージを処理します。

```
118 # TODO: 【Step.3】 7.CloudroboticsApiの受信メッセージから会話理解APIのもののみを取得する
119 elif received_message.header['AppId'] == '<...>' and received_message.header['AppProcessingId'] == '<...>':
120     if received_message.body['success'] == 'true' or received_message.body['type'] == 'finishTalk':
121         if received_message.body['type'] == 'init' or len(received_message.body['type']) >= 4 and received_mess
122             # TODO: 【Step.3】 8.AIからの応答メッセージの取得
123             visitor = '<...>'
124             visitor_id = '<...>'
125             self.type = '<...>'
126             talkByAi = '<...>'
127             for talk in talkByAi:
128                 self.onSuccess('<...>')
```

・ visitor/visitor_idには任意の値を設定してください。

・ self.type/talkByAiには、API仕様書を参考に会話理解APIの受信メッセージから適切な値を設定してください。

・ onSuccessの引数に、会話理解APIの受信メッセージのtalkByAiからSayTextの値を設定してください。

課題: 3

- ・ 会話理解APIから受信したメッセージを処理します。

```
131 # TODO: 【Step.3】 9.AIからの終了メッセージ取得後の処理
132 self.type = received_message.body['type'] = 'init'
133 talkByAi = received_message.body['talkByAi']
134 for talk in talkByAi:
135     self.onSuccess('<...>')
```

- ・ onSuccessの引数に、会話理解APIの受信メッセージのtalkByAiからSayTextの値を設定してください。



Appendix

サンプルコードの説明

ハンズオンで提供されるファイルは以下のとおりです。

Choregrapheプロジェクトファイル群			説明
cloudrobotics api_handson 02	lib	cloudrobotics	Cloud Robotics APIモジュール
		translator	翻訳API用
		conversation	会話理解API用
		paho	MQTT通信モジュール
	sample1		翻訳APIサンプル用ビヘイビア
	sample2		会話理解APIサンプル用ビヘイビア
	handson2_1		翻訳API ハンズオン用 課題ビヘイビア
	handson2_1_ans		翻訳APIハンズオン用 サンプルビヘイビア
	handson2_2		会話理解API ハンズオン用 課題ビヘイビア
	handson2_2_ans		会話理解APIハンズオン用 サンプルビヘイビア

