

ORACLE®

Oracle Digital Assistant

The Complete Training

Webview component

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Topic agenda

- 1 ➤ Introduction to Webview
- 2 ➤ Webview architecture
- 3 ➤ How to call a Webview
- 4 ➤ Demo

Topic agenda

1 ➤ Introduction to Webview

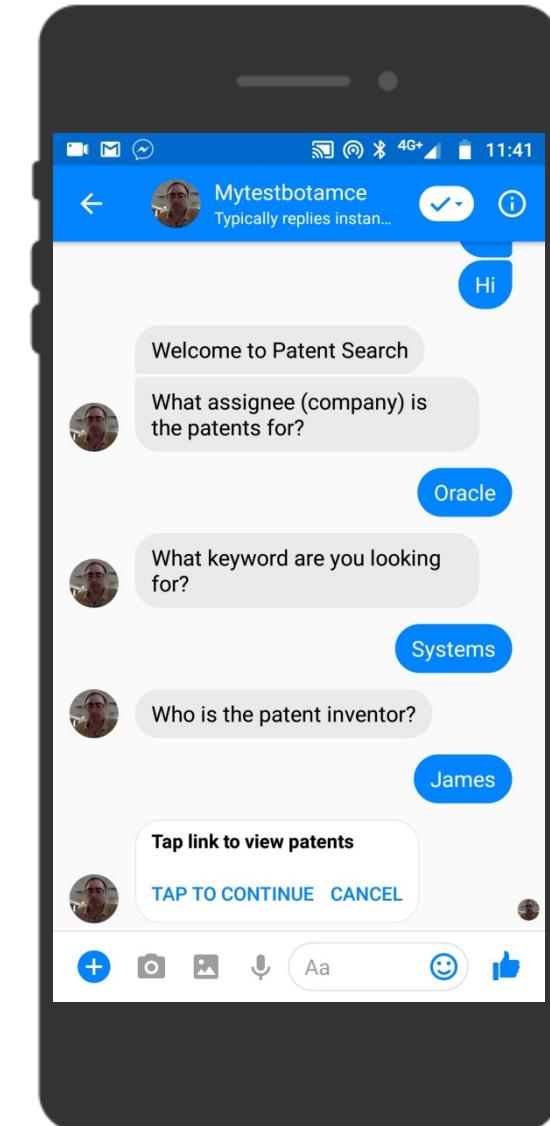
2 ➤ Webview architecture

3 ➤ How to call a Webview

4 ➤ Demo

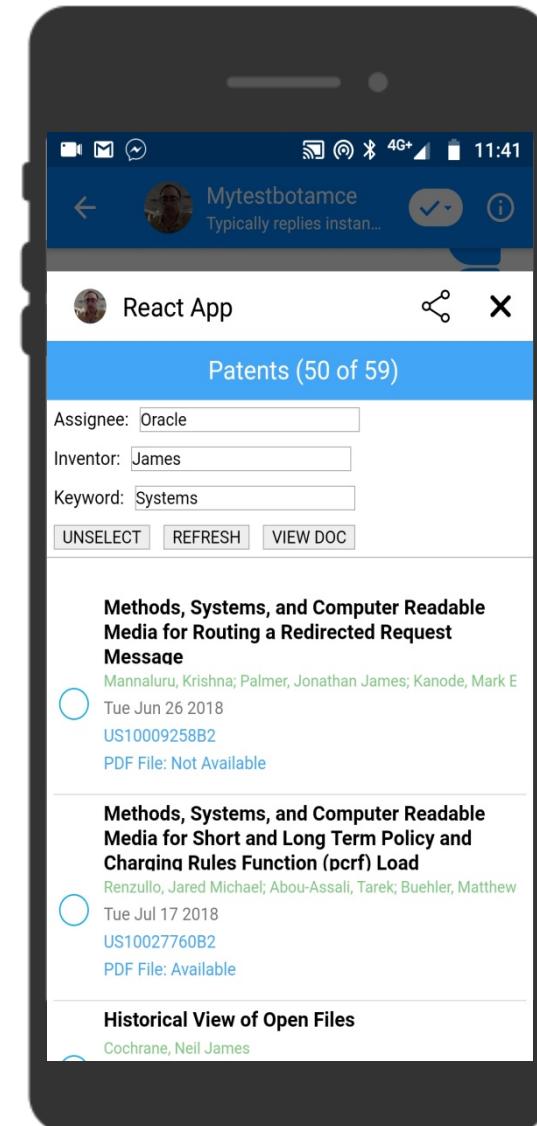
A typical conversation

- Natural language conversation
 - Unstructured
 - Free flowing
 - “Conversational”
- But not all conversation
 - Credit card details
 - Delivery address
 - Scanning a voucher/coupon
 - Viewing and selecting from multiple options

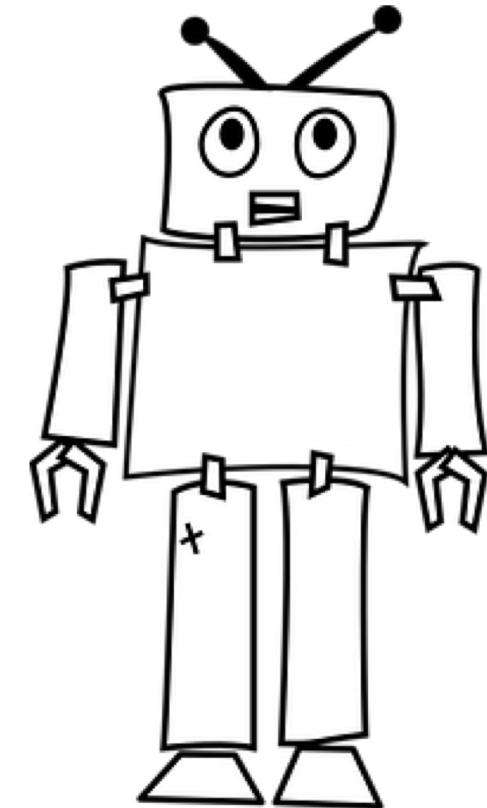


A typical conversation

- Some parts of a conversation are better handled in a structured way
 - User knows exactly what information is required
 - Discrete choices
 - Faster input
 - Immediate validation of input
 - Information capture
 - Upload image
 - Signature
 - Scan barcode



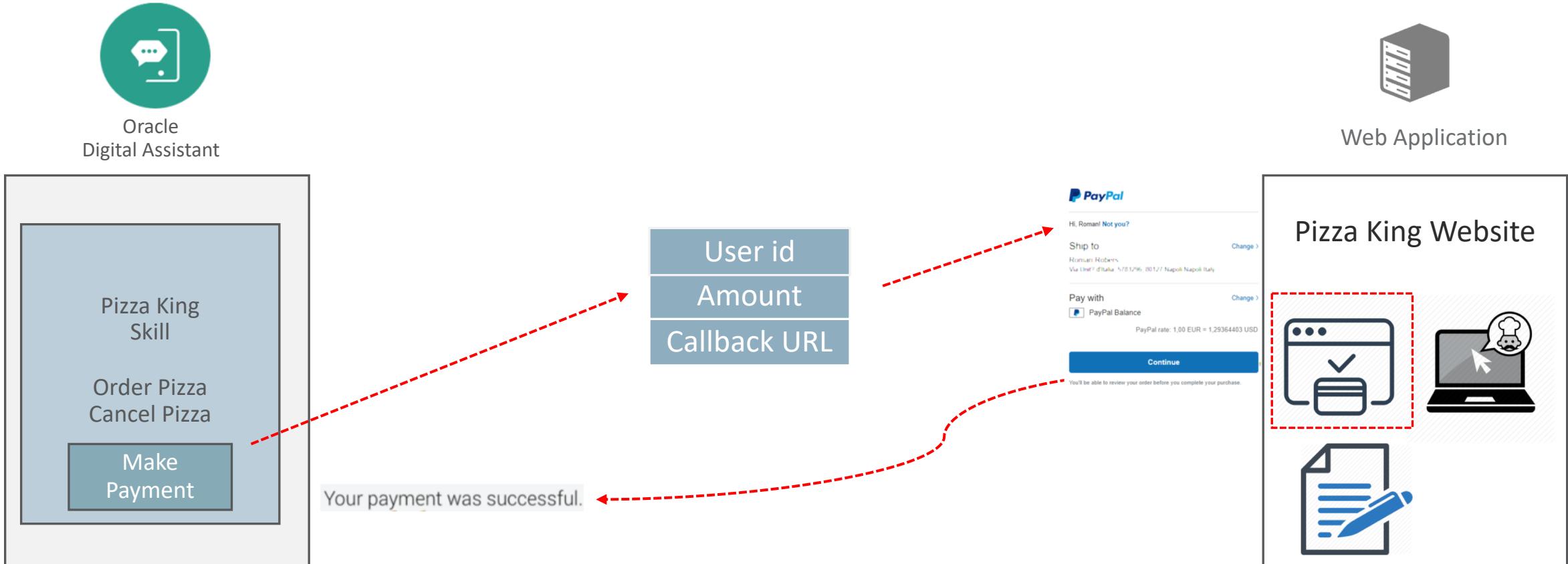
The System.Webview component
integrates **external web application**
calls to a bot conversation



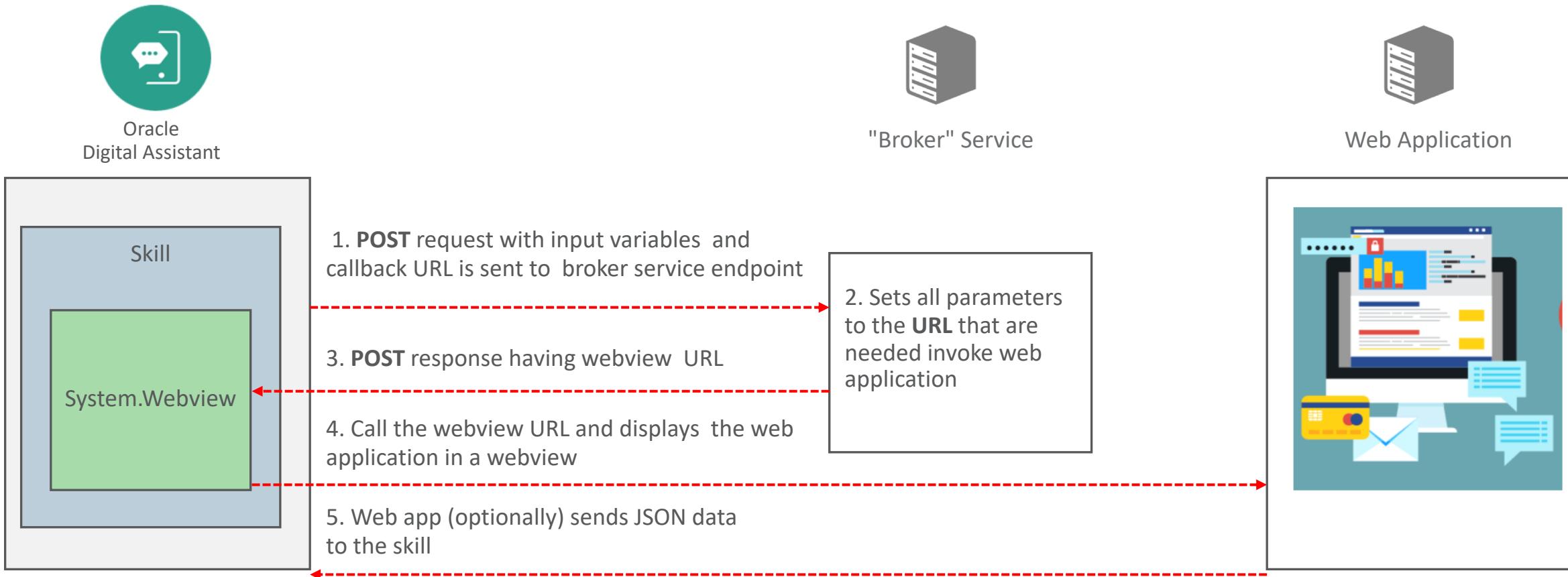
Topic agenda

- 1 ➤ Introduction to Webview
- 2 ➤ Webview architecture
- 3 ➤ How to call a Webview
- 4 ➤ Demo

Use case



How the WebView works



Topic agenda

- 1 ➤ Introduction to Webview
- 2 ➤ Webview architecture
- 3 ➤ How to call a Webview
- 4 ➤ Demo

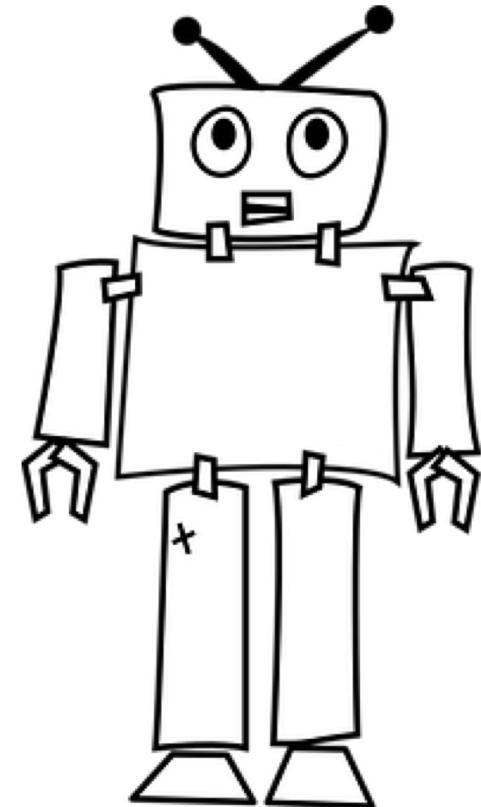
ODA: WebView OBotML

```
9 context:  
10 variables:  
11   firstname: "string"  
12   outputfromweb: "string"  
13  
14 states:  
15  
16 askFirstName:  
17   component: "System.Text"  
18   properties:  
19     prompt: "What is your first name?"  
20     variable: "firstname"  
21  
22 webview:  
23   component: "System.Webview"  
24   properties:  
25     sourceVariableList: "firstname"  
26     variable: "outputfromweb" → System.Webview component  
27     prompt: "Please tap on the link to proceed"  
28     webAppUrl: "https://testsystemwebview.herokuapp.com/webviewparams" → Source variable list  
29   transitions: {}  
30  
31 output:  
32   component: "System.Output"  
33   properties:  
34     text: "${outputfromweb.value.pdfURL}" → Variable to store callback URL JSON payload  
35     keepTurn: false  
36     translate:  
37   transitions:  
38     return: "done" → "POST" request endpoint Base URL
```

System WebView component properties

Name	Description	Required
sourceVariableList	A comma-separated list of context or user variable names	Yes
variable	To store webview's callback payload	Yes
prompt	A string to be shown as the text of the link to access the webview	No
webAppUrl	The base URL to which the values of the parameters on the sourceVariableList will be sent	Yes
authToken	The authorization token to be sent with requests to the webAppUrl	No
queryParams	URL query parameters key-value pairs to be appended to the webAppUrl.	No

The parameter **webview.onDone** is automatically added to the request payload to pass the **callback URL** property



Request payload sent from System.Webview

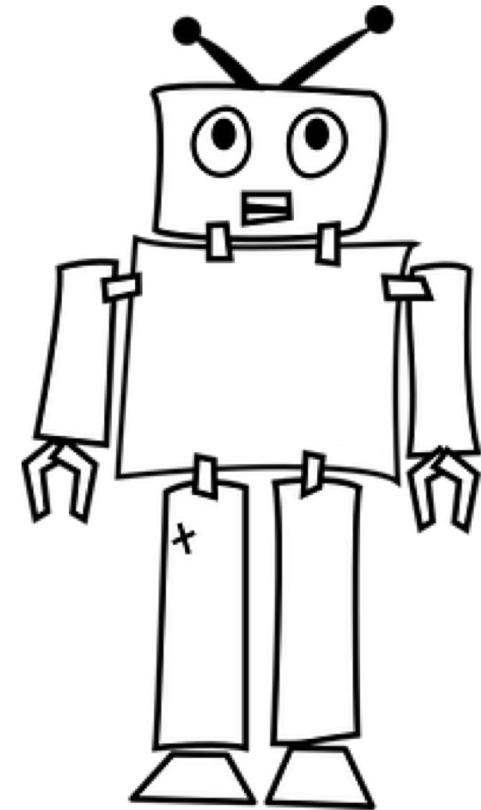
```
1 {  
2   "parameters": [  
3     {"value": "rohit",  
4      "key": "firstname"  
5    }, {  
6      "value": "https://bot.x.oraclecloud.com:443/connectors/v1/callback?state=e502e373-44e3-4c2c-a3e7-13776d5fcce4",  
7      "key": "webview.onDone"  
8    }]  
9 }
```

"Broker" service response payload

- The response from the "broker service" must be a JSON payload.
- The payload must consist of a single property **webview.url** with the link to access the web application to display in the webview UI.
- This **webview.url** is used by the `System.Webview` component to display the web application

```
{  
  "webview.url": "http://01620dae.ngrok.io/webviewApp/d93ad3a8-4c38-4fc6-87dd-39957125bafc/index.html"  
}
```

The Web app is launched in an
embedded WebView



Web application response upon completion

- Web application needs make a POST request to the provided callback URL
 - Pass results back to System.Webview component
- Request body must JSON format containing return values

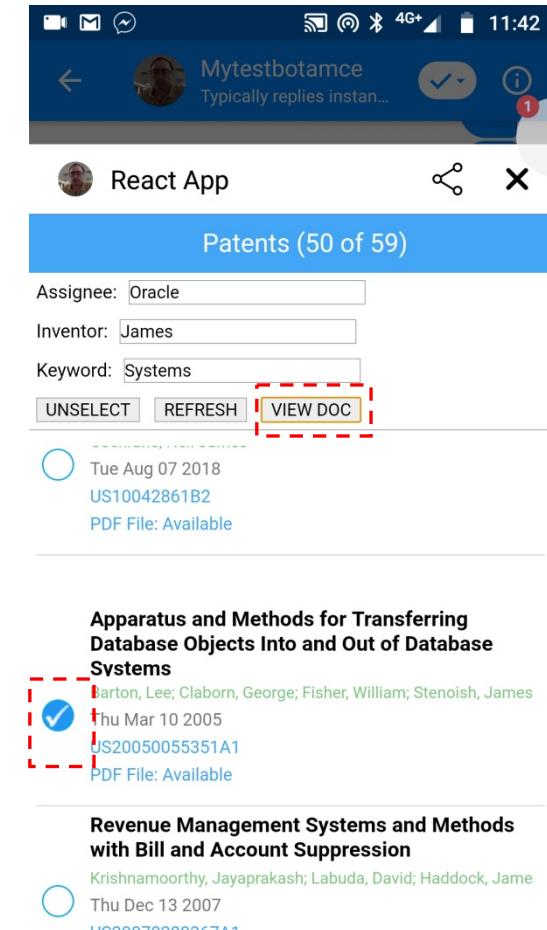
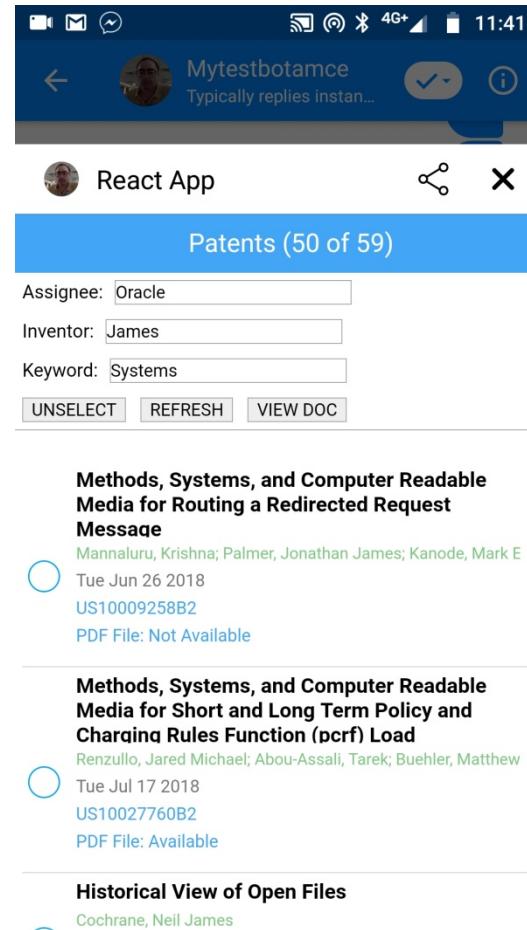
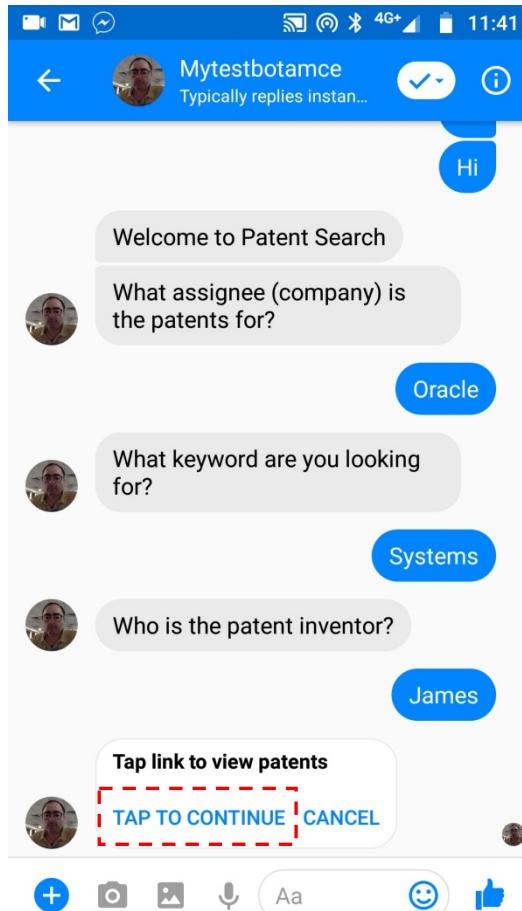
```
1 var settings = {  
2   "async": true,  
3   "crossDomain": true,  
4   "url": "https://01620dae.ngrok.io/connectors/v1/callback?state=73b5266b-8ae7-4652  
      -a21c-b0baa99b0ed2",  
5   "method": "POST",  
6   "headers": {  
7     "Content-Type": "application/json"  
8   },  
9   "data": "{\"pdfURL\":\"http://www.africau.edu/images/default/sample.pdf\"}"  
10 }  
11  
12 $.ajax(settings).done(function (response) {  
13   console.log(response);  
14 });|
```

Callback URL

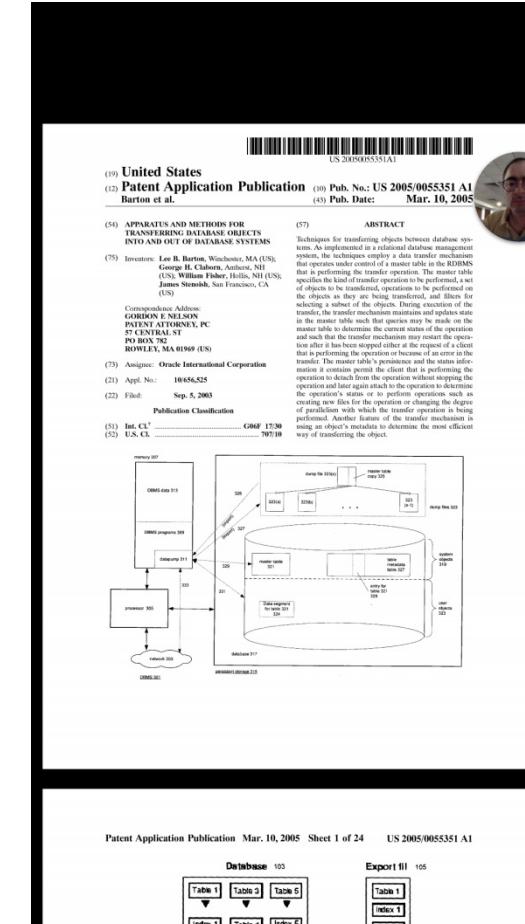
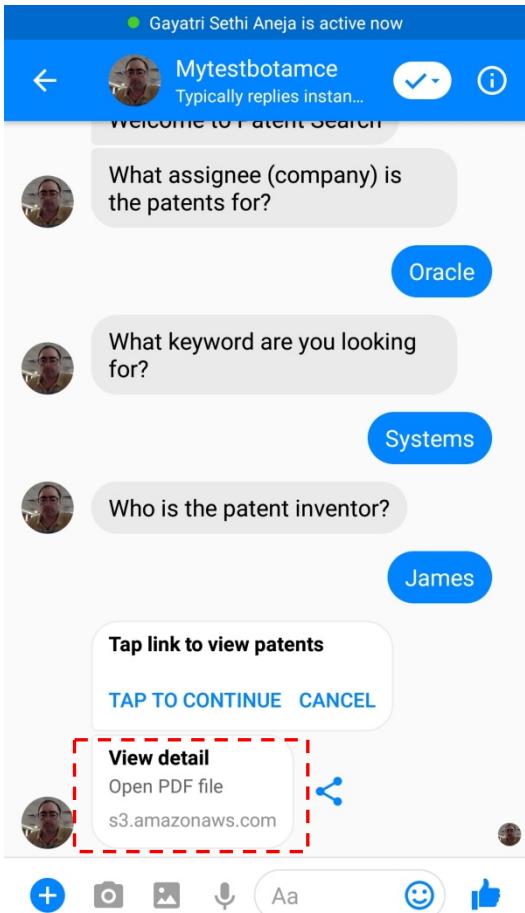
Topic agenda

- 1 ➤ Introduction to Webview
- 2 ➤ Webview architecture
- 3 ➤ How to use
- 4 ➤ Demo

Demo



Demo



Integrated Cloud Applications & Platform Services



Oracle Digital Assistant Hands-On

ORACLE®