
ORACLE®

Oracle Digital Assistant

The Complete Training

Webview component

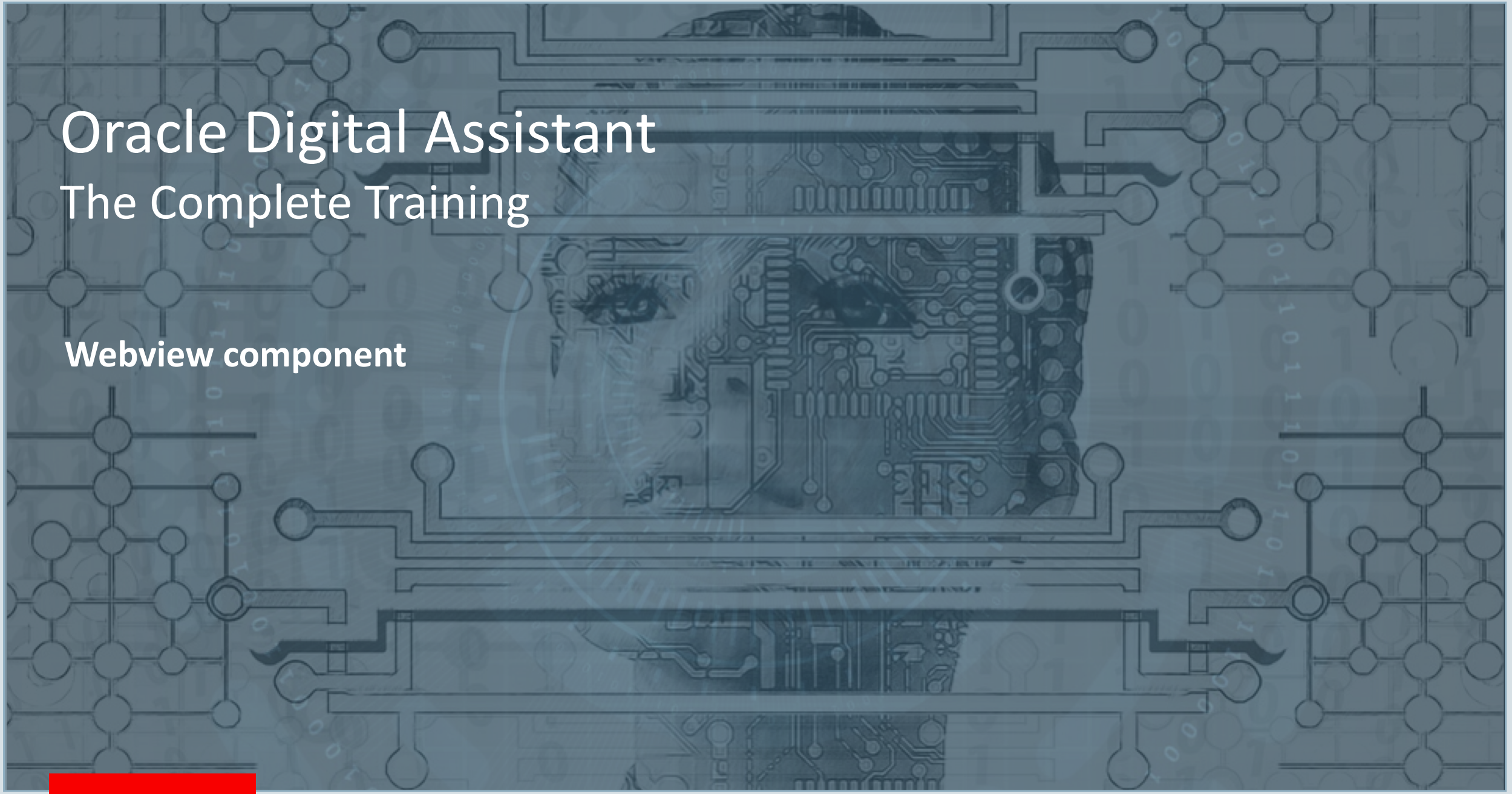


Image courtesy of pixabay.com

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

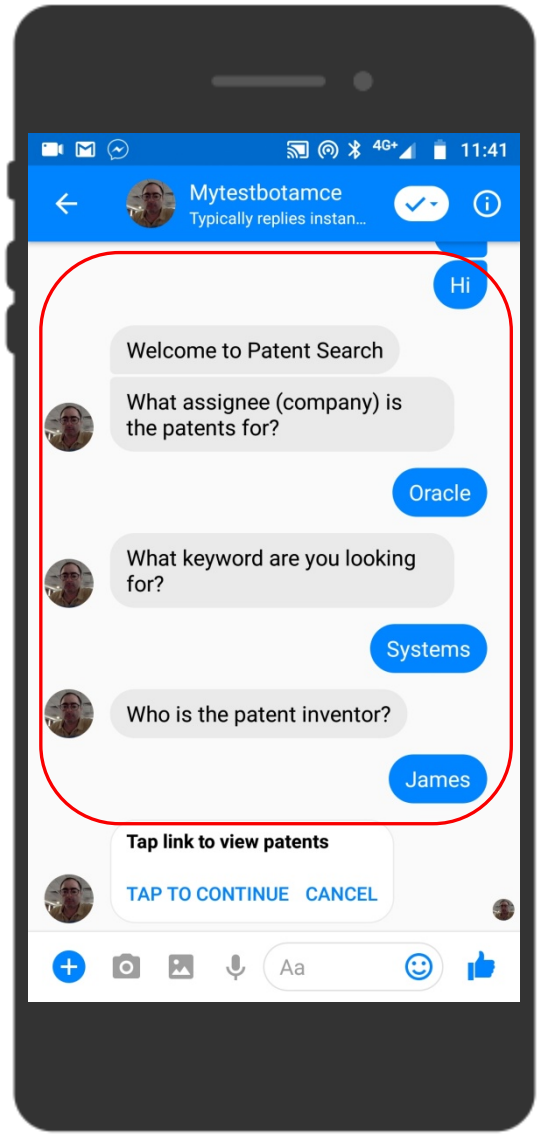
1. Introduction to Webview
2. Architecture of Webview in ODA
3. How to call a Webview
4. Demo

Topic agenda

1. Introduction to Webview
2. Architecture of Webview in ODA
3. How to call a Webview
4. Demo

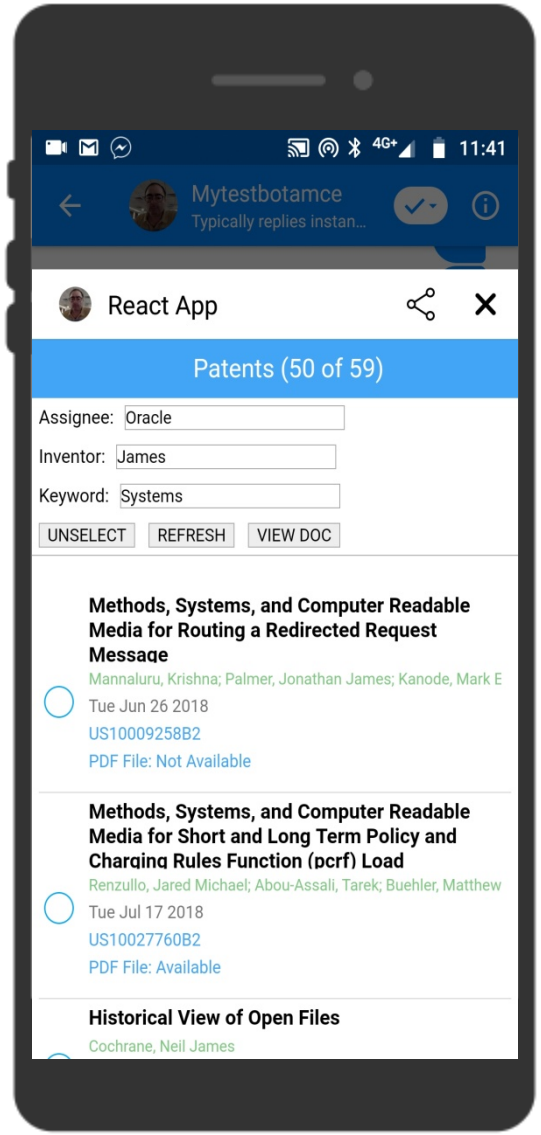
A typical conversation

- 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

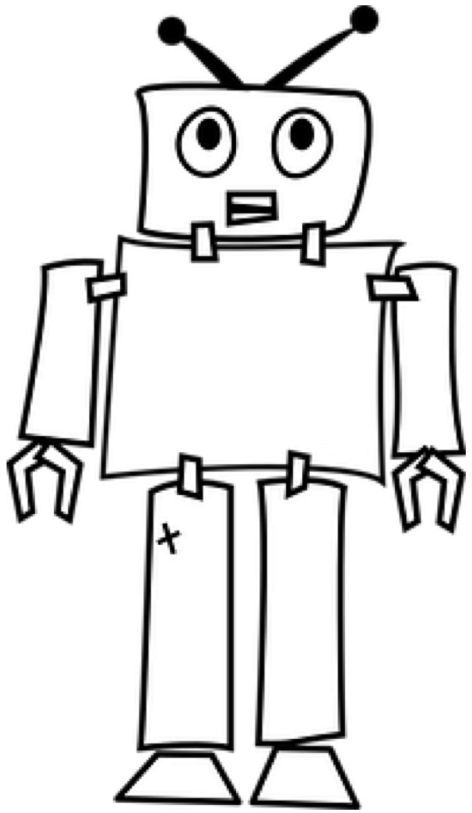


A typical conversation

- 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



System WebView component allows you to **access an external web app**



Instant Apps vs. WebView

Instant App

- Built-in technology
- Visual layout editor to build pre-defined layouts and page flows
- Declarative action handling

WebView

- Your programming language and tool of choice
- Easier integration of existing web forms
- Allows dynamic forms (fields created at runtime)
- Host the web app anywhere

Topic agenda

1. Introduction to Webview
2. Architecture of Webview in ODA
3. How to call a Webview
4. Demo

Use case

Oracle
Digital Assistant

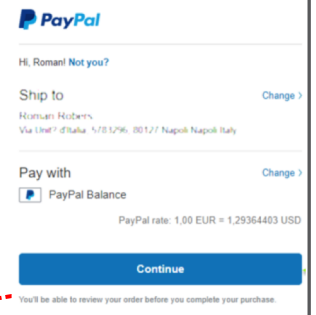
Web Application

Pizza King
Skill

Customize Pizza
Order Pizza
Cancel Pizza

Make
Payment

User id
Amount
Callback URL



Pizza King Website

Your payment was successful.

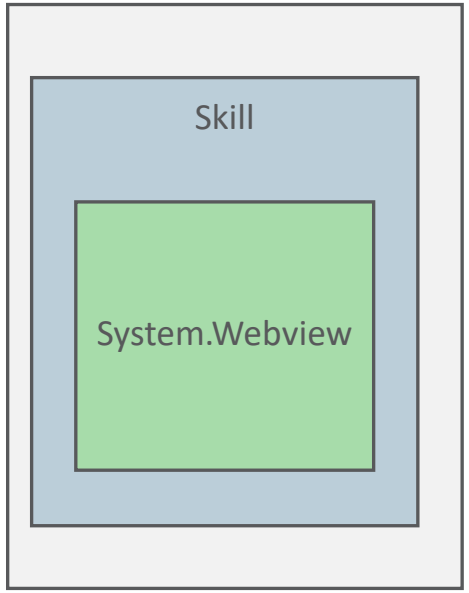
Your order is almost ready.

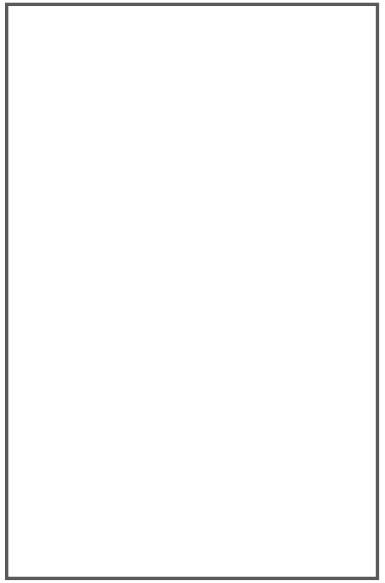
How the WebView works

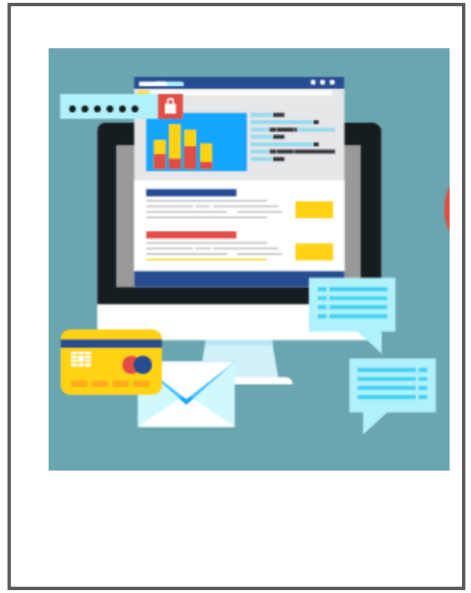
Oracle
Digital Assistant

Web Server

Web Application





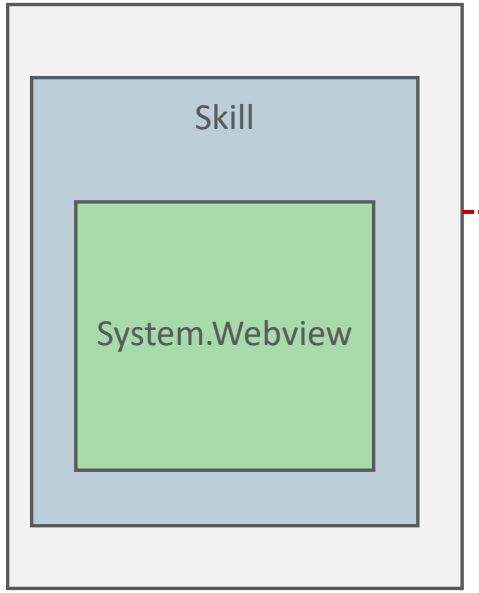


How the WebView works

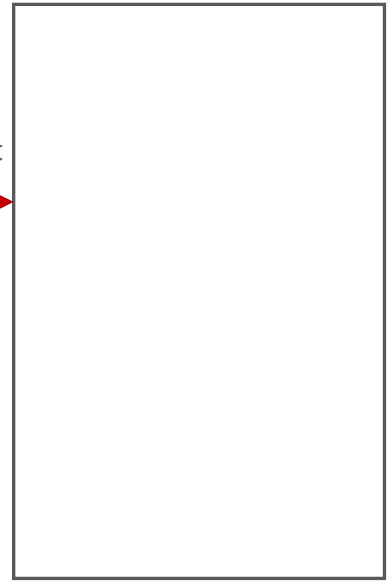
Oracle
Digital Assistant

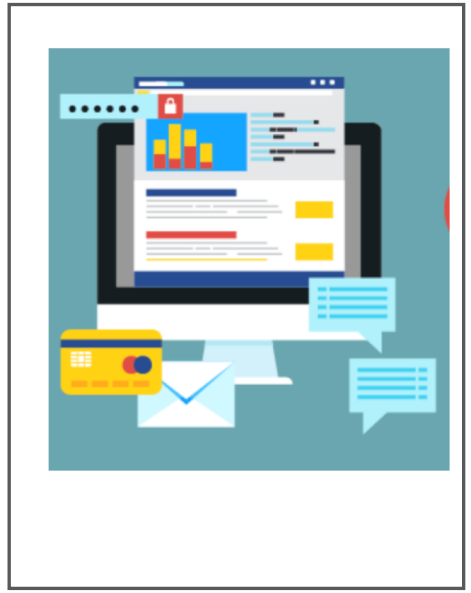
Web Server

Web Application



1. POST request with input variables and callback URL is sent to webserver endpoint



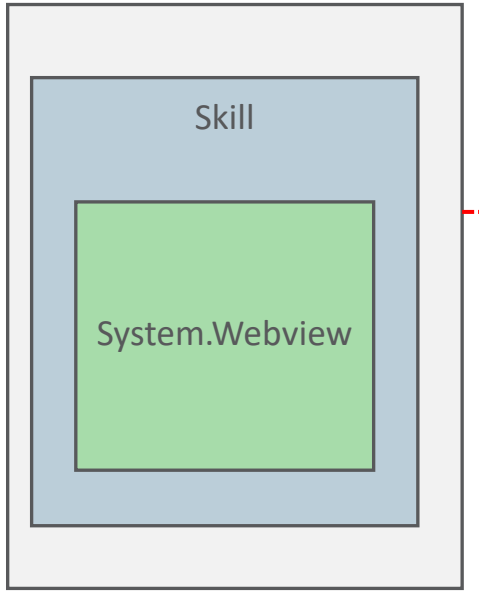


How the WebView works

Oracle
Digital Assistant

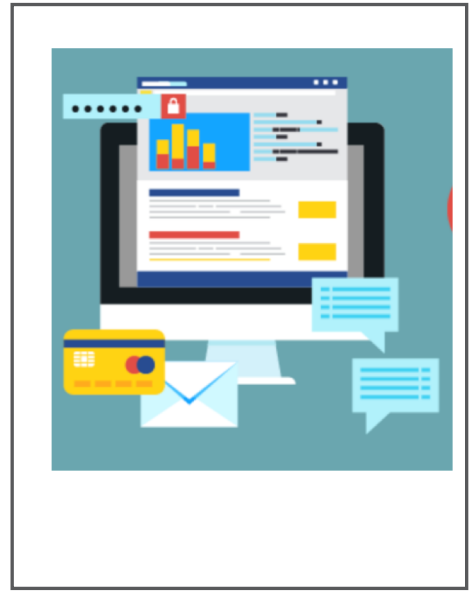
Web Server

Web Application

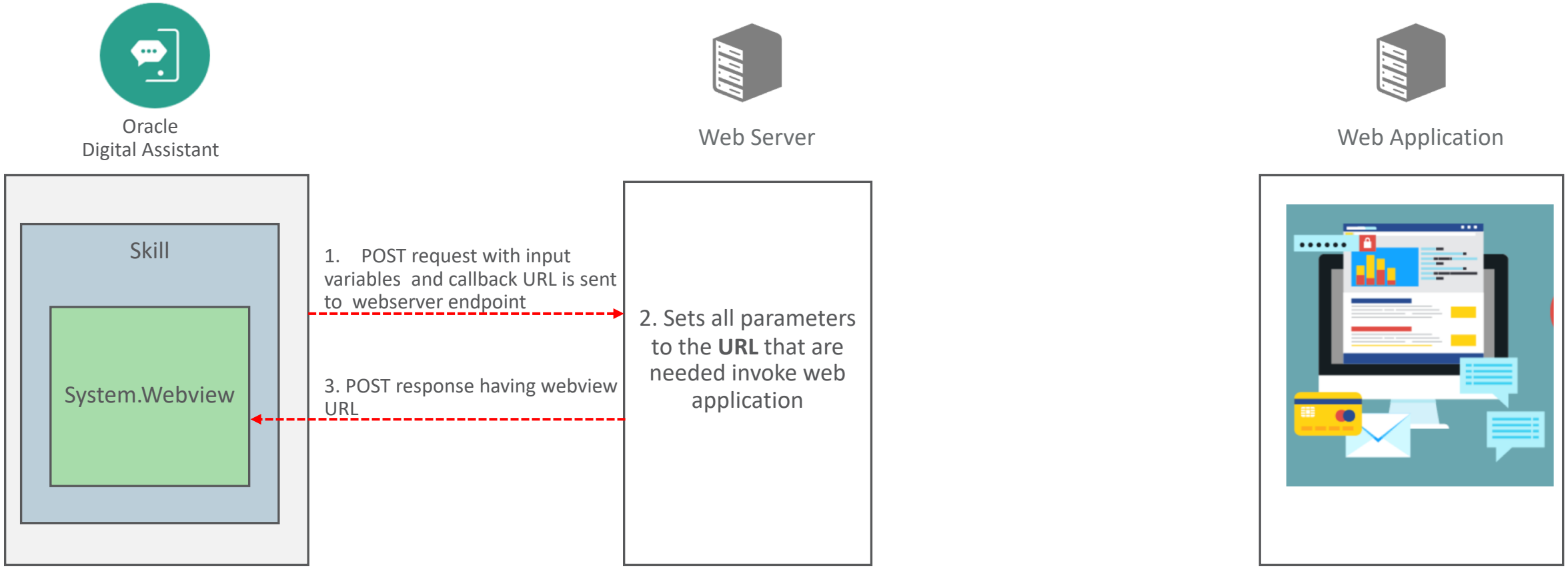


1. POST request with input variables and callback URL is sent to webserver endpoint

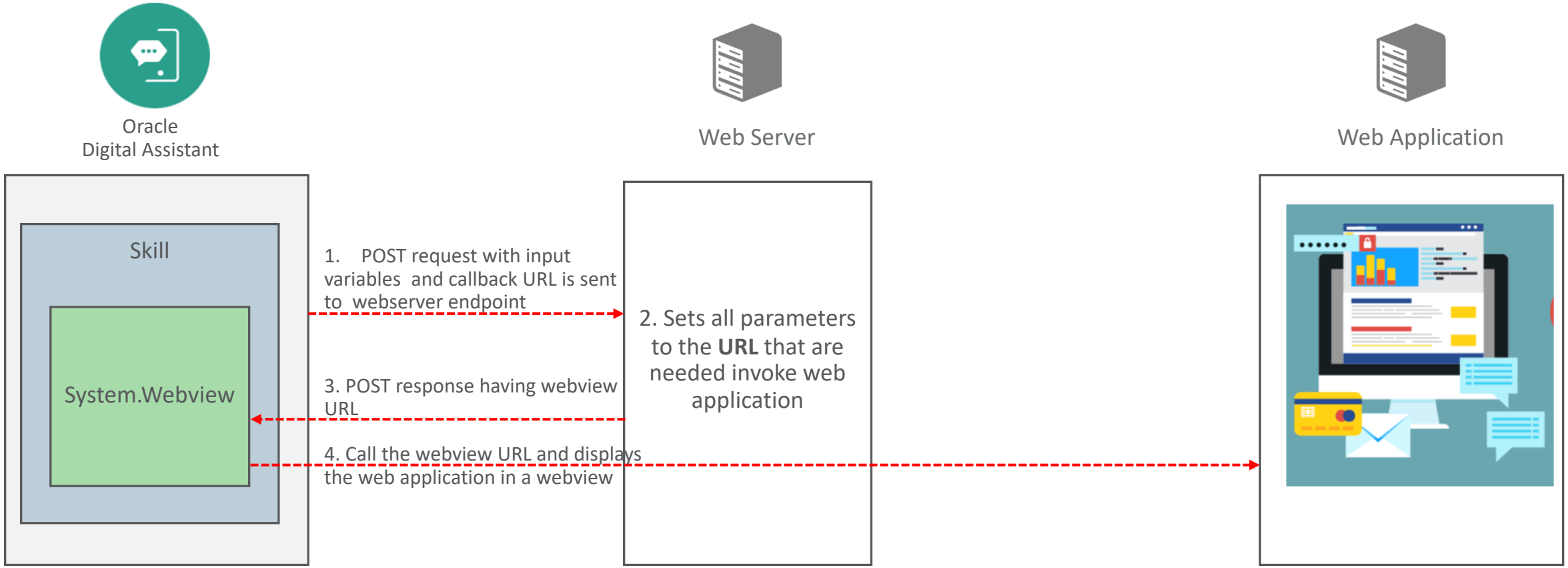
2. Sets all parameters to the **URL** that are needed invoke web application



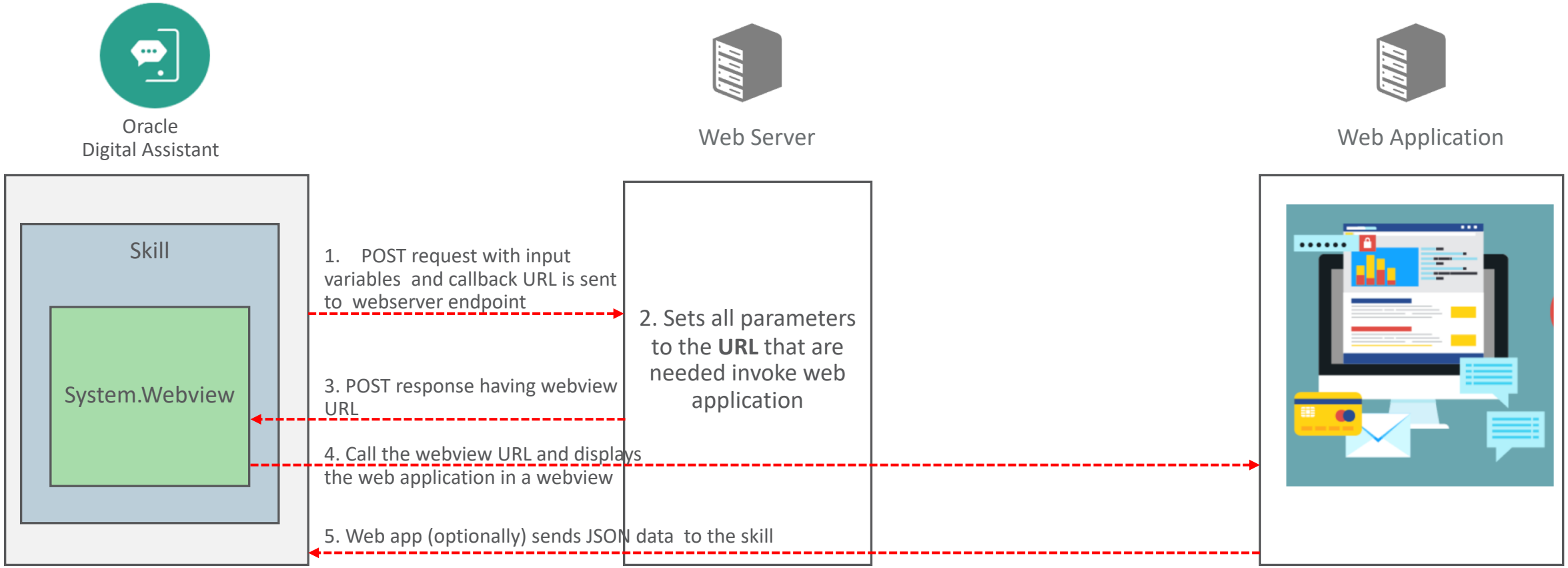
How the WebView works



How the WebView works



How the WebView works



Summary - how the WebView works

- System.WebView makes POST request with input variable and callback URL to webserver
- WebServer sets all parameters to the URL needed to invoke web app
- POST response having Webview URL to access the web application is sent back to the ODA
- ODA makes a Get call to the URL and displays the web application in webview
- User accesses the application, and before returning to ODA can send JSON data to the ODA

Topic agenda

1. Introduction to Webview
2. Architecture of Webview in ODA
3. How to call a Webview
4. Demo

ODA: WebView OBotML

```
context:
  variables:
    firstname: "string"
    outputfromweb: "string"

states:

  askFirstName:
    component: "System.Text"
    properties:
     prompt: "What is your first name?"
     variable: "firstname"

  webview:
    component: "System.Webview"
    properties:
      sourceVariableList: "firstname"
      variable: "outputfromweb"
      prompt: "Please tap on the link to proceed"
      webAppUrl: "https://testsystemwebview.herokuapp.com/webviewparams"
    transitions: {}

  output:
    component: "System.Output"
    properties:
      text: "${outputfromweb.value.pdfURL}"
      keepTurn: false
      translate:
    transitions:
      return: "done"
```

System WebView component

Source variable list

Variable to store callback URL JSON payload

Post 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**" will be automatically added to the request payload to pass on the **callback URL** property

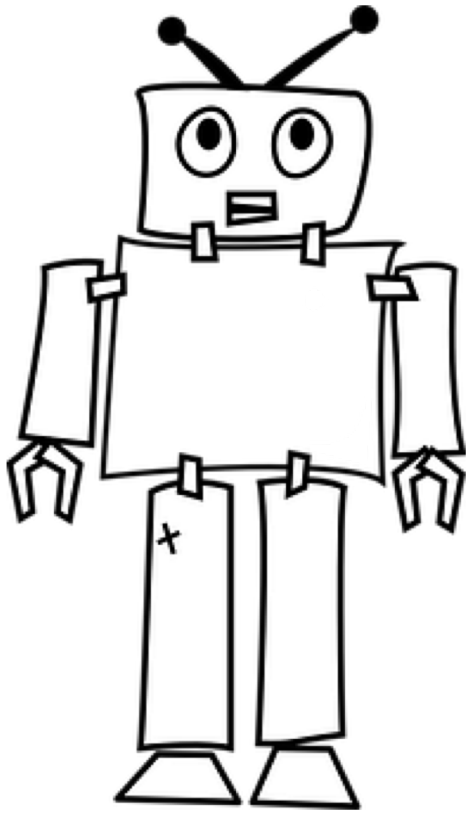


Image courtesy of pixabay.com

Request payload sent from System.WebView

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

Automatically added to the request payload

WebServer: POST response payload

- The response from the POST request to the web app endpoint should be a JSON payload.
- The payload must have a single property "webview.url" with the link to access the webview UI.
- This webview.url downloads to page in 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

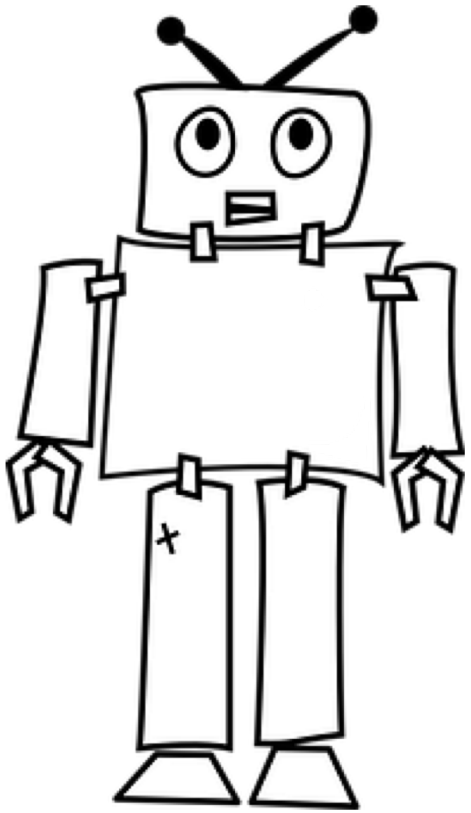


Image courtesy of pixabay.com

Web Application: Response on completion

- The web application needs make a POST request to send back response to the DA
  - Request URL will be the callback URL
  - Request body will be in JSON format containing the values to pass back to Bots.

```
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 });
```

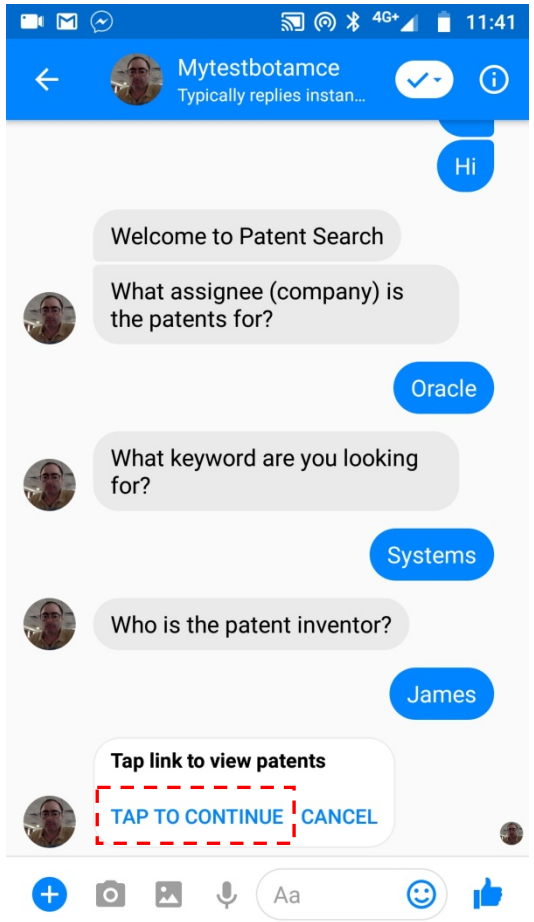
Callback URL

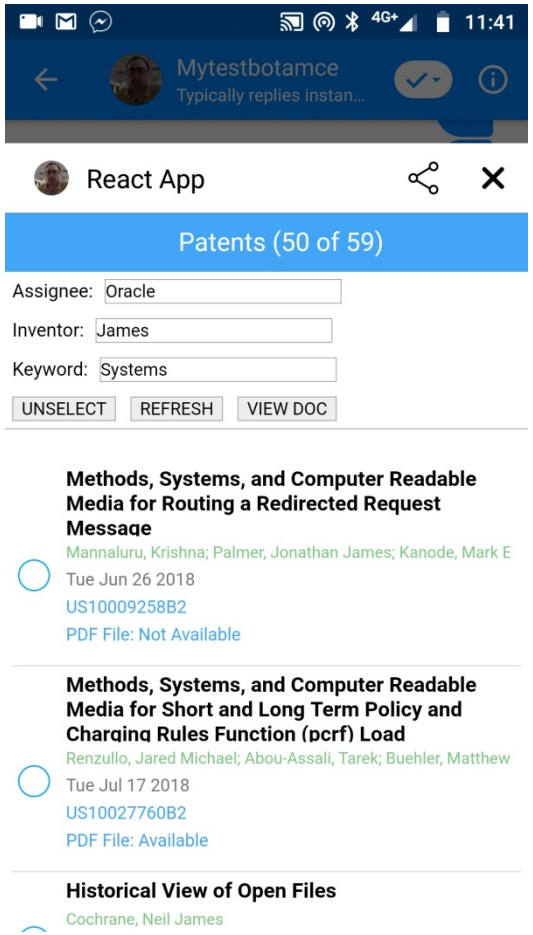
JSON Payload

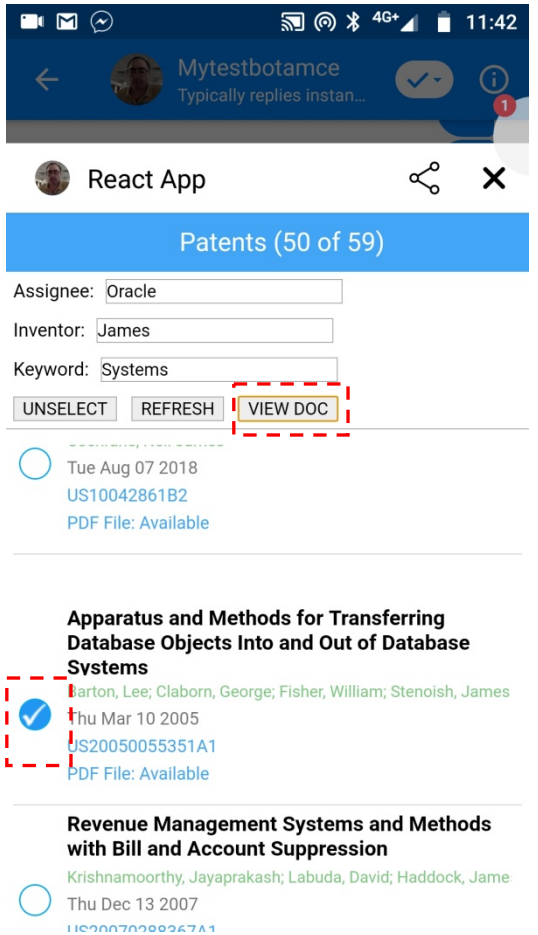
Topic agenda

1. Introduction to Webview
2. Architecture of Webview in ODA
3. How to use
4. Demo

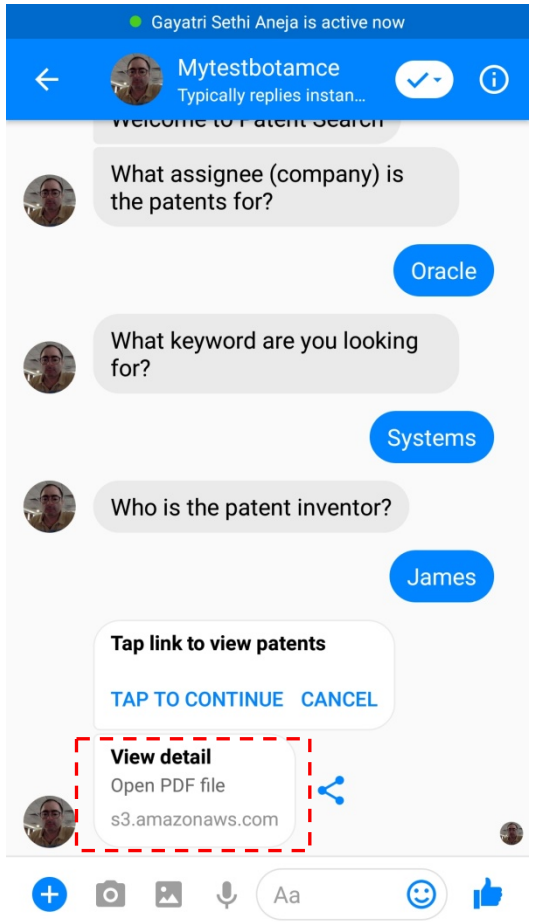
Demo

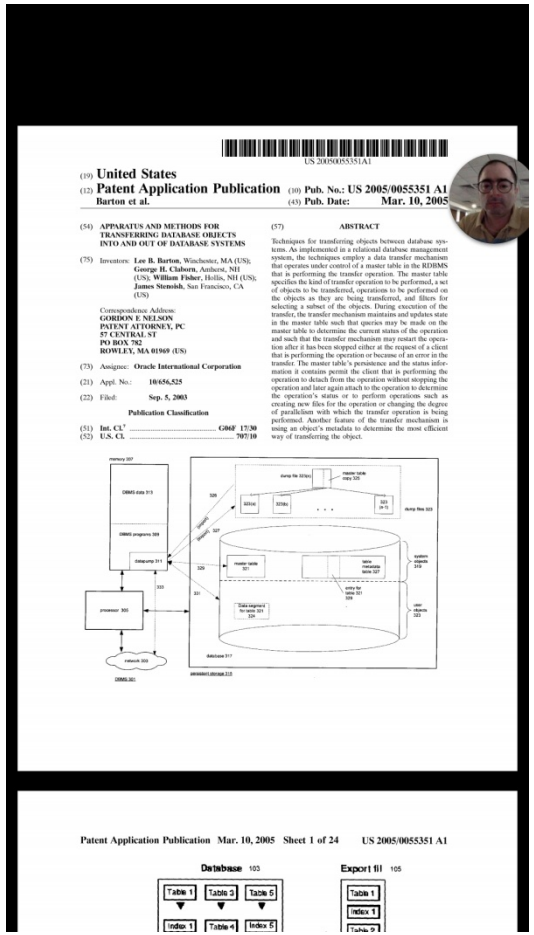






Demo





Oracle Digital Assistant Hands-On

TBD

Integrated Cloud

Applications & Platform Services


ORACLE®