

ODTUG  
Kscope19   
SEATTLE, WASHINGTON • JUNE 23-27

PLEASE FILL OUT  
YOUR EVALUATIONS

Jon Dixon > Handling SMS Messages  
with APEX, ORDS, and Twillio

SEATTLE

 Washington State  
Convention Center

# Agenda

- Text Messaging & Twillio
- Case Study
- Demo SMS Survey App
- Demo Deep Dive
- Key Takeaways
- Q&A

Handling SMS Messages with APEX, ORDS, and Twillio

# About Me



# About Me

- Working with Oracle technology since 1994
- 18 Years with Oracle consulting
- The past 6 years
  - Extending Oracle ERP with APEX & ORDS
  - Adding cloud superpowers to APEX (API Gateway, Lambda, S3, Twillio)
- Co-Founder JMJC Cloud
  - On-Premise & Cloud ERP extensions using APEX & ORDS
  - APEX and native mobile apps
  - Integrations solutions using ORDS, OIC, Dell Boomi etc.



Handling SMS Messages with APEX, ORDS, and Twillio

# About SMS & Twillio

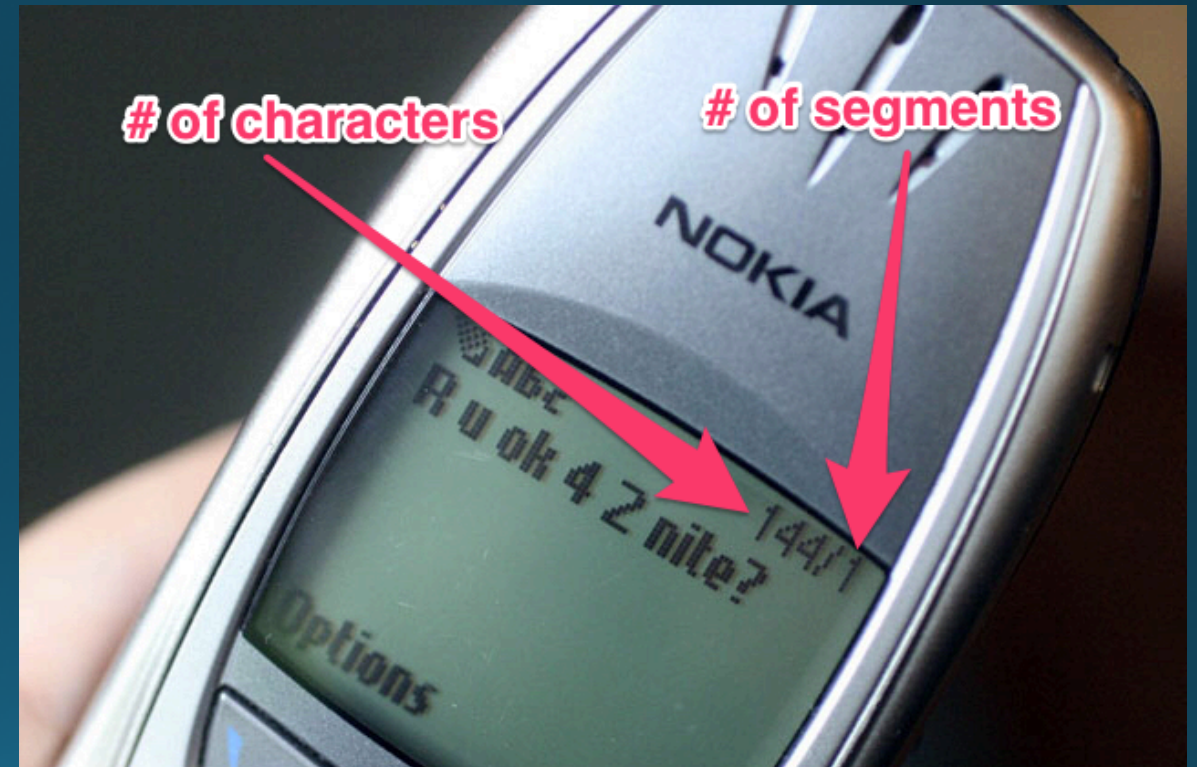
# About Text Messages

- The first text message
  - “merry Christmas”
  - Sent in the UK December 3, 1992
- Today (2017)
  - 81% US population text  
(Intelligent Contacts)
  - Worldwide, 16m messages sent / minute  
(World Economic Forum)



# About Text Messages

- Max Size
  - 1,600 characters
  - 160 character segments
- Segments
  - How the SMS industry & Twilio count messages



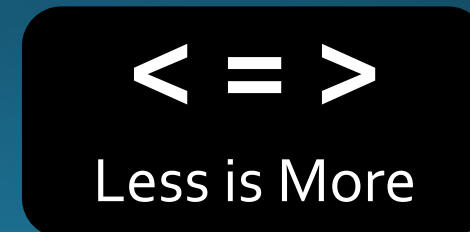


# Why Text Messages?

- Immediacy
  - 99% Open Rate (SinglePoint)
  - 95% Open < 3 Mins (Forbes)

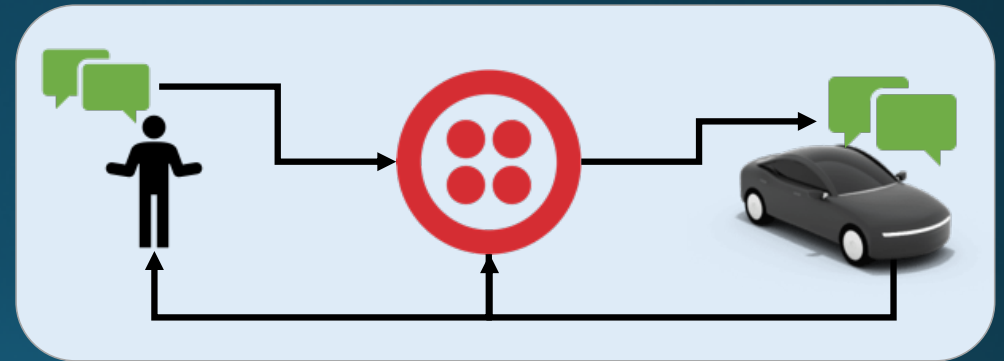
- Mobility

- Brevity



# About Twilio

- Offers text message brokering
- Used by Uber, Lyft, Airbnb etc.
- Near real time
- Delivery confirmation (callback)
- Easy to use REST APIs
- Send & receive text and images
- \$0.0075 to send / receive a SMS




# Setting up Twillio

- Create an Account
- Create Project
  - \$15 trial credit outbound only
  - Upgrade project
- Note Account SID & Auth Token
- Configure the Webhook
- Buy a phone number

PROJECT NAME	KSCOPE 2019	<a href="#">edit</a>
ACCOUNT SID	AC64da20f4902741f13282c0daf7fa26aa	
AUTH TOKEN		<a href="#">view</a>

## Buy a Number



United States

(+1)

▼

Number

▼

512









Capabilities

▼

Search Term Match: First part of number

Type: All

Requirement: Any

NUMBER	TYPE	CAPABILITIES				PRICE
		VOICE	SMS	MMS	FAX	
<div><div>+1 (512) 647-2461</div><div>AUSTIN, TX</div></div>	Local					<div><div>\$1.00</div><div>monthly</div></div>
<div><div>+1 (512) 361-5351</div><div>BUDA, TX</div></div>	Local					<div><div>\$1.00</div><div>monthly</div></div>



# Twilio Outbound API

- Send messages to cell numbers
- Test with API Explorer in Twilio
- Username = Account SID, Password = Auth Token

<https://api.twilio.com/2010-04-01/Accounts/AC7237b4eec0e9160aea5930d26222/Messages.json>

The screenshot shows the Twilio API Explorer interface. The 'Authorization' tab is selected, displaying the 'Basic Auth' type. The 'Username' field contains the Account SID 'AC7237b4eec0e9160aea5930d26c05c81b', and the 'Password' field contains the Auth Token. Below this, the 'Body' tab is selected, showing the request body configuration. The 'Body' is set to 'Hello World'. The 'From' field is '+15124025682' and the 'To' field is '+13303100591'. The 'Content-Type' is set to 'x-www-form-urlencoded'.

KEY	VALUE
Body	Hello World
From	+15124025682
To	+13303100591

POST



The screenshot shows the Twilio API Explorer interface displaying the JSON response. The 'JSON' tab is selected, showing the response body. The response is a JSON object with the following fields: 'sid', 'date\_created', 'date\_updated', 'date\_sent', 'account\_sid', 'to', 'from', 'messaging\_service\_sid', 'body', 'status', and 'num\_segments'.

```
1 - {
2   "sid": "SMebed53ec70e54dfefb6148ba85a98f1b4",
3   "date_created": "Sat, 18 May 2019 14:26:23 +0000",
4   "date_updated": "Sat, 18 May 2019 14:26:23 +0000",
5   "date_sent": null,
6   "account_sid": "AC7237b4eec0e9160aea5930d26c05c81b",
7   "to": "+13303100591",
8   "from": "+15124025682",
9   "messaging_service_sid": null,
10  "body": "Hello World",
11  "status": "queued",
12  "num_segments": "1",
13  "num_bytes": "14"
14 }
```

# Twilio Webhook

- Twilio calls the Webhook for every inbound message
- You can have a different Webhook for each phone number
- Use to handle responses from people
- Point to your **internet accessible** ORDS web service

Messaging

CONFIGURE WITH	Webhooks, TwiML Bins, Functions, Studio, or Proxy ▾		
A MESSAGE COMES IN	Webhook ▾	https://[redacted]com/ords/xxords/twilio/webhook	HTTP POST ▾
PRIMARY HANDLER FAILS	Webhook ▾		HTTP POST ▾

Handling SMS Messages with APEX, ORDS, and Twillio

# Case Study

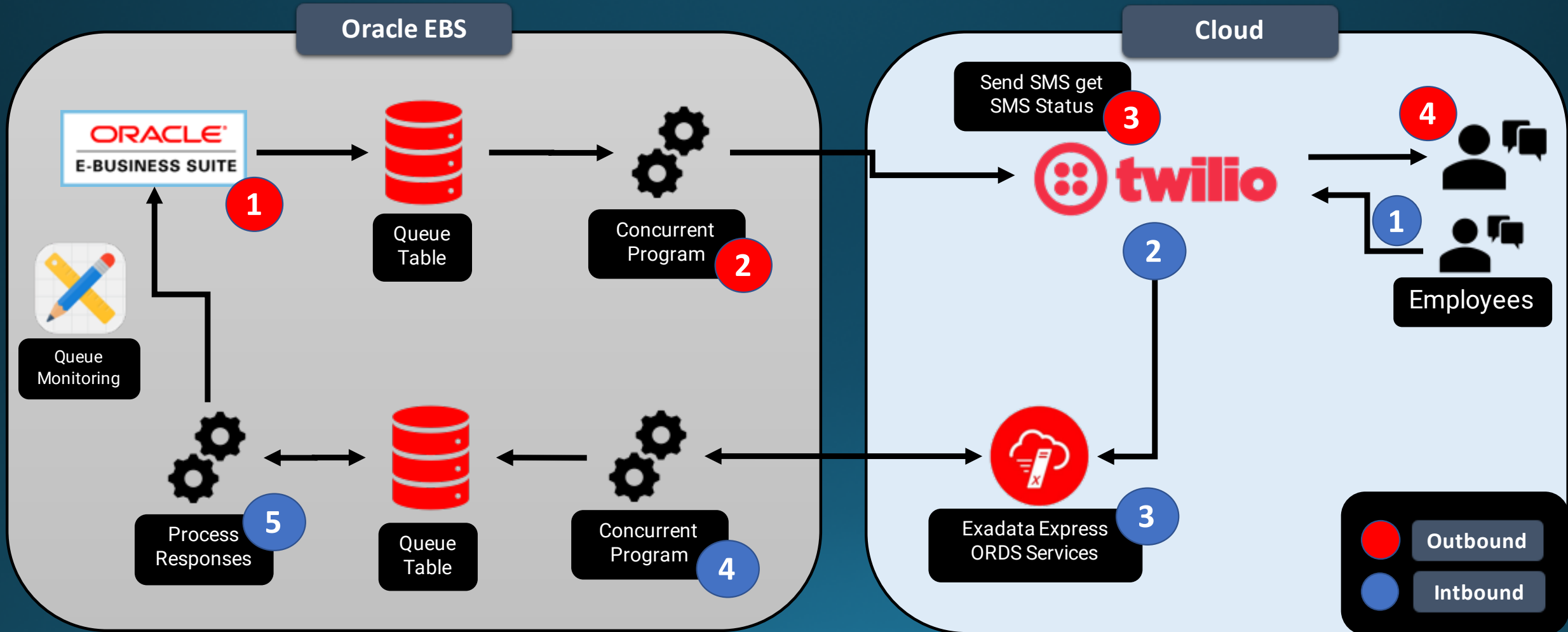


# Welder Certification Expirations

- Skilled welders
- No certification no work
- SMS reminders sent at intervals prior to certification expiration
- Responses visible to managers
- Utilize E-Business Suite, APEX & ORDS, Exadata Express & Twillio



# Architecture



# EBS & APEX Configurations

- Mobile numbers stored on HR record
- Certs & dates stored in HR EIT attributes
- Simplified maintenance via an APEX App
- EBS Lookups to store configurations
- Custom tables for queued messages
- Concurrent programs to
  - Queue messages
  - Send queued messages to Twillio
  - Pull in responses from Exadata Express





# Lessons Learned

## When to send an SMS?



# Lessons Learned

## Configure a test number



# Lessons Learned

## Data quality

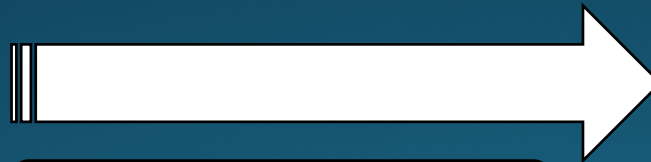
- Correct phone number
- Format standardization



(330) 310 0591

330-310-0591

3303100591



**E.164 Format**

[+] [country code] [subscriber number including area code]

**+13303100591**

# Lessons Learned

## International numbers



+1

+44

...



# Lessons Learned

## Error notifications



# Lessons Learned

## Handling Conversations



# Lessons Learned

Who sent that text?



# Lessons Learned

## Opting Out



Handling SMS Messages with APEX, ORDS, and Twillio

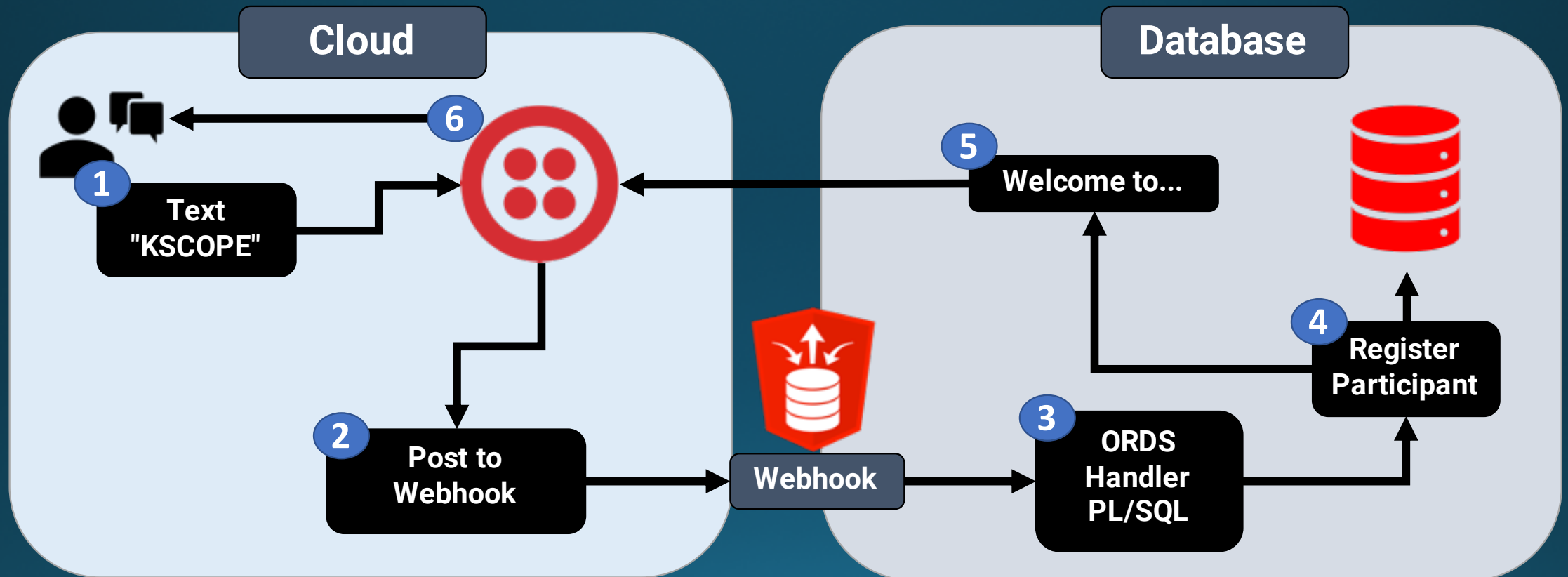
# Demo – SMS Poll



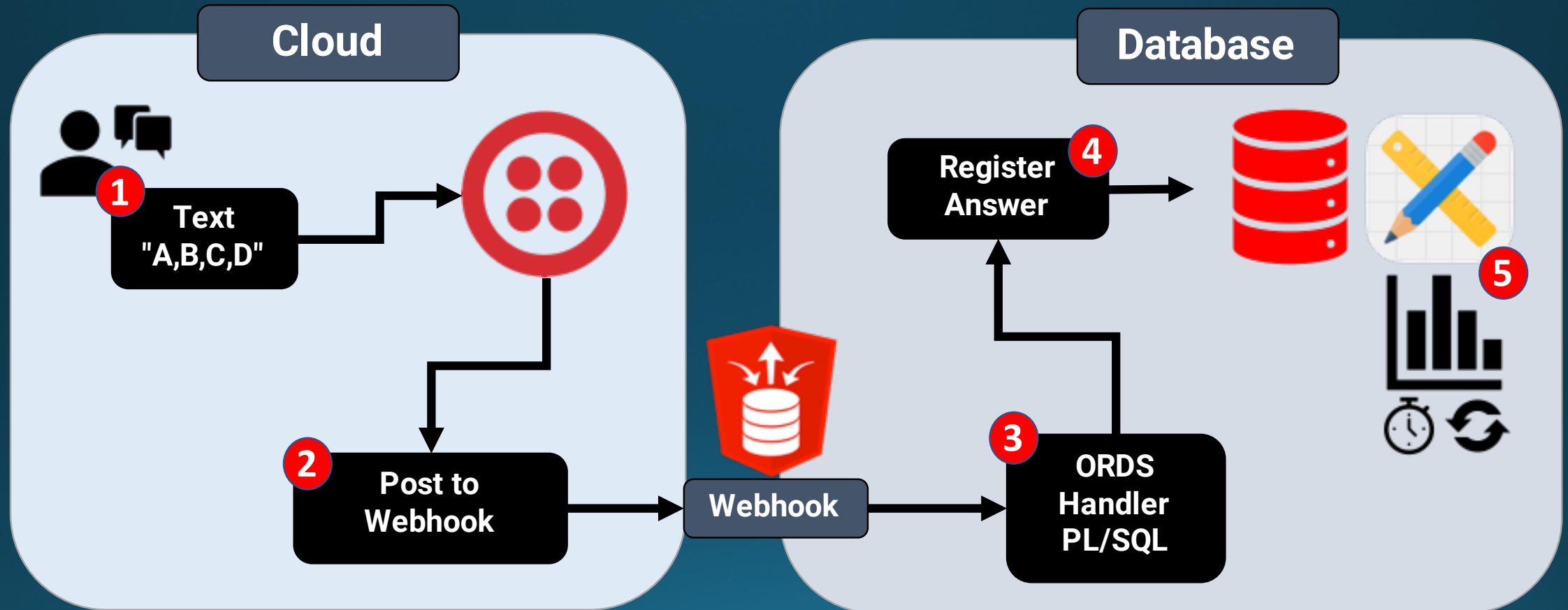
Handling SMS Messages with APEX, ORDS, and Twillio

# Demo - Deep Dive

# Demo Flow - Signup



# Demo Flow – Answer Question



# Oracle REST Data Services (ORDS)

- Host REST based web services
- Build using SQL and PL/SQL
- Auto REST (1 click web service)
- REST Enabled SQL
- Secure via OAuth2
- Used as listener for APEX
- Installable independently of APEX

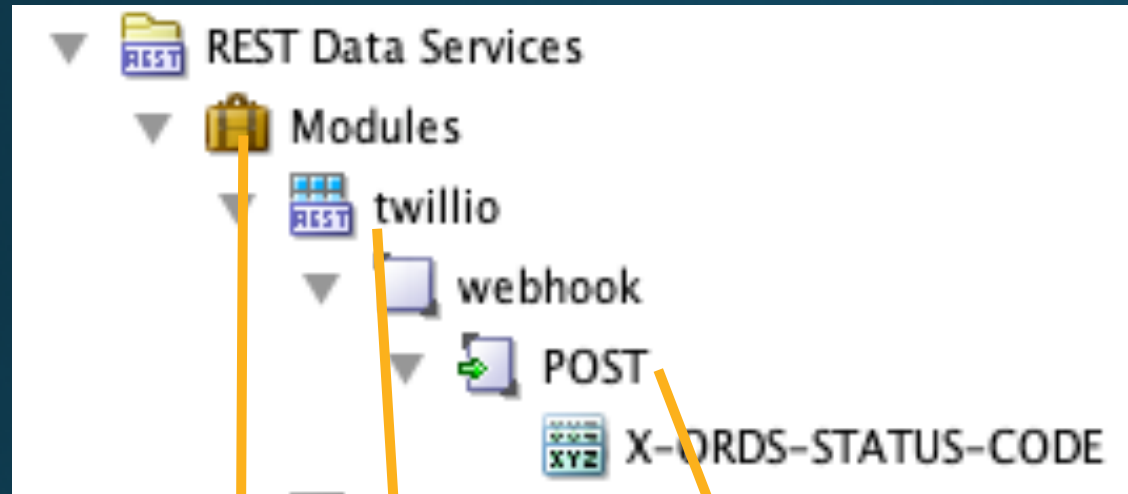


# ORDS Handler (Webhook)

- Receive message payload from Twillio
- Call procedure to parse payload
- Call procedure to perform polling logic
  - Register user & send welcome SMS
  - Insert poll response into responses table



# ORDS Webhook Definition



Module

Template

Handler

```
DECLARE
  l_payload_clob CLOB;
BEGIN

  -- Get payload from Twillio
  l_payload_clob := :body_text;

  jd.sms_utl_pk.twillio_web_hook
    (p_payload      => l_payload_clob,
     p_channel_code => 'POLL');

  :http_status_code := 201;

EXCEPTION WHEN OTHERS THEN
  :http_status_code := 400;
END;
```

# Twilio Payload

```
ToCountry=US&ToState=TX&SmsMessageSid=SMde480a5dec7cb666ec9fb1d593dbaa89&NumMedia=0&ToCity=AUSTIN&FromZip=33186&SmsSid=SMde480a5dec7cb666ec9fb1d593dbaa89&FromState=FL&SmsStatus=received&FromCity=M  
IAMI&Body=Hello%20World&FromCountry=US&To=%2B1512  
4025682&ToZip=78736&NumSegments=1&MessageSid=SMde  
480a5dec7cb666ec9fb1d593dbaa89&AccountSid=AC7237b  
4eec0e9160aea5930d26c05c81b&From=%2B13303100591&A  
piVersion=2010-04-01
```

Response

Sent from  
Number

Sent to  
Number

# Parse Twillio Payload

```
lt_payload := apex_string.split(l_msg_body, '&');  
  
FOR i IN 1..lt_payload.COUNT LOOP  
    l_param_str := lt_payload(i);  
    l_param := SUBSTR(l_param_str, 1, INSTR(l_param_str, '=', 1) - 1);  
    l_value := SUBSTR(l_param_str, INSTR(l_param_str, '=', 1) + 1);  
  
    CASE l_param  
        WHEN 'From' THEN  
            lr_twillio_ib_rec.from_number := UTL_URL.UNESCAPE (l_value);  
        WHEN 'To' THEN  
            lr_twillio_ib_rec.to_number := UTL_URL.UNESCAPE (l_value);  
        WHEN 'Body' THEN  
            lr_twillio_ib_rec.message_body :=  
                REPLACE(UTL_URL.UNESCAPE(l_value), '+', ' ');  
        WHEN 'SmsMessageSid' THEN  
            lr_twillio_ib_rec.sender_ref := UTL_URL.UNESCAPE (l_value);  
        ELSE NULL;  
    END CASE;  
END LOOP;
```

Create Array of  
Name Value Pairs

Extract key fields from  
name value pairs

```
INSERT INTO sms_inbound  
VALUES lr_twillio_ib_rec;  
  
poll_utl.webhook_subscription  
(pr_twillio_ib_rec => lr_twillio_ib_rec);
```

Log details to table & call  
API to perform poll logic

# Call Twillio Messages API

```
GC_ACCOUNT          CONSTANT VARCHAR2(100) := 'AC7237b4eec0e9160aea5930d26112233';
GC_SMS_URL          CONSTANT VARCHAR2(500) := 'https://api.twilio.com/2010-04-01/Accounts/';
```

```
PROCEDURE send_sms
(p_from_number IN VARCHAR2,
 p_to_number   IN VARCHAR2,
 p_msg         IN VARCHAR2) IS

  l_sms_url      VARCHAR2(500) := GC_SMS_URL ||
                                GC_ACCOUNT ||
                                '/Messages.json';

  l_post_body    VARCHAR2(500);
  l_clob         CLOB;
```

Build Twillio  
API URL

Prepare and Call  
Twillio Web Service

```
-- Set HTTP Headers for APEX Web Service Call
apex_web_service.g_request_headers.DELETE();
apex_web_service.g_request_headers(1).name := 'Content-Type';
apex_web_service.g_request_headers(1).value :=
  'application/x-www-form-urlencoded';

-- Prepare Message Body
l_post_body := 'Body=' || utl_url.escape(p_msg, TRUE) ||
  '&From=' || utl_url.escape(p_from_number, TRUE) ||
  '&To=' || utl_url.escape(p_to_number, TRUE);

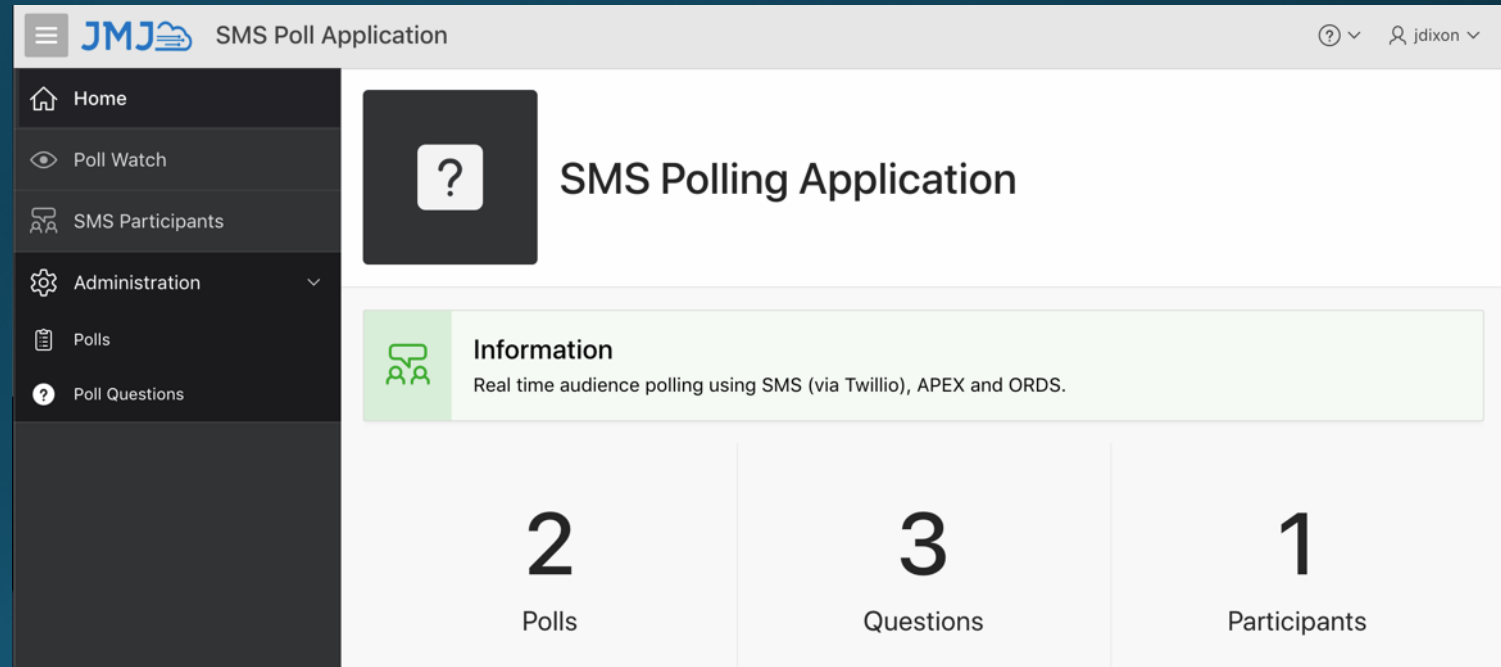
-- Call Twillio API
l_clob := apex_web_service.make_rest_request(
  p_url      => l_sms_url,
  p_http_method => 'POST',
  p_body      => l_post_body,
  p_username  => GC_ACCOUNT,
  p_password  => GC_PASSWORD);

IF apex_web_service.g_status_code = 201 THEN
  APEX_JSON.PARSE(l_clob, true);
...

```

# APEX Application

- 5 Pages
  - Maintain Polls
  - Maintain Questions
  - Enroll Participants
  - Run Poll
  - SMS Participants





Handling SMS Messages with APEX, ORDS, and Twillio

# Key Takeaways

# Key Takeaways

- Buy a Twillio number per 'Topic'
- Install Twillio SSL certificate (TLS 1.2) in DB wallet
- ORDS webhook service must be accessible from the Internet
- The recipient won't know who SMS came from
- Don't send text messages at midnight
- Work on data quality, realize it will never be perfect

Jon Dixon

[jon.dixon@jmjcloud.com](mailto:jon.dixon@jmjcloud.com)

[www.jmjcloud.com](http://www.jmjcloud.com)

[@jmjcloud](#)

Handling SMS Messages with APEX, ORDS, and Twillio

# Thank You!



ODTUG  
Kscope19   
SEATTLE, WASHINGTON • JUNE 23-27

**PLEASE FILL OUT  
YOUR EVALUATIONS**

SEATTLE

 Washington State  
Convention Center