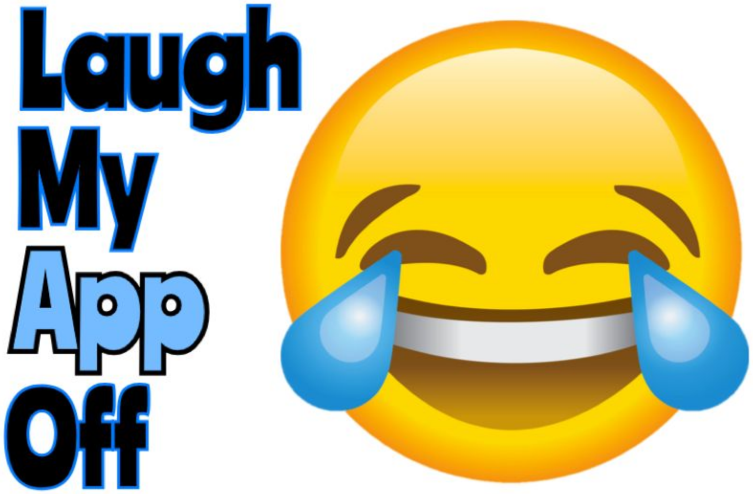
AWS Security Project: Laugh My App Off

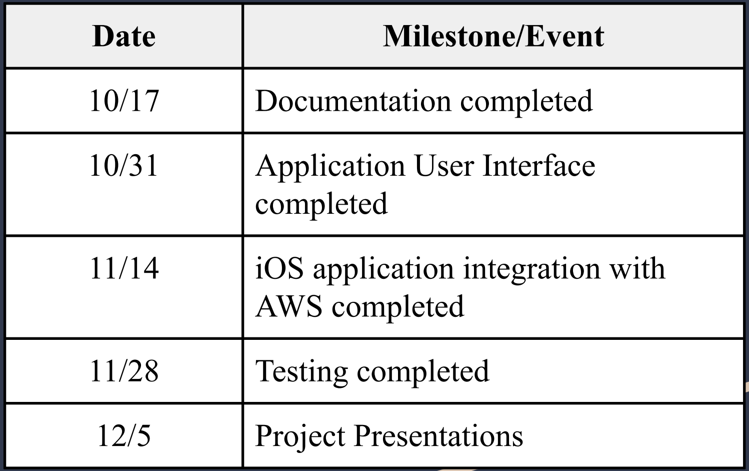
Paul Bundac
KerryAnn DeMeester
Megan Rapach

Functionality

- "Joke of the day" application
  - SMS text of new joke, same time everyday
- iOS app
  - Register
  - Log in
  - View all jokes in database



Planning



Date	Milestone/Event
10/17	Documentation completed
10/31	Application User Interface completed
11/14	iOS application integration with AWS completed
11/28	Testing completed
12/5	Project Presentations

- Project Manager: Megan Rapach
- Critical Success Factor: **project planning**
    - Assignment = design & build secure app
    - Planning with security in mind at every phase of SDLC
- Project Scope
    - Features/Functionality of App
        - Secure sign-in and user registration
        - Display daily jokes from DB
        - Daily SMS texts
    - AWS
        - Fetch jokes via API
        - Store jokes in database
        - Securely store user info
        - Trigger SMS text to iOS device

Security Risks – Overview

Protecting user data in AWS

- Phone Number
- First Name
- Username
- Password

Ensuring secure login and user registration

- iOS biometrics = PII*
- Interfacing between mobile app & AWS

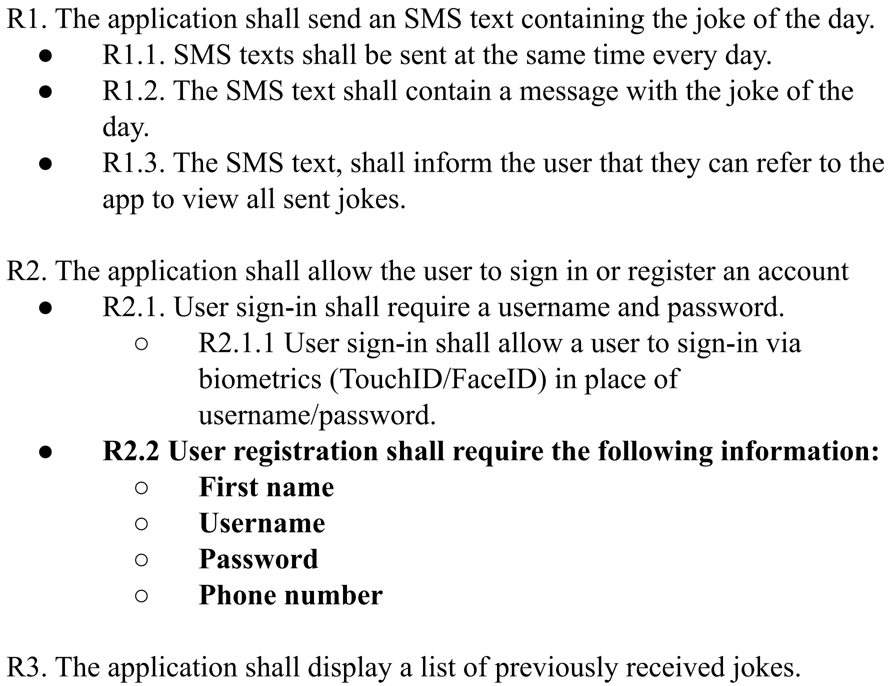
Ensuring secure app deployment

- Generate .ipa file - iOS App Store Package
- Archive app from XCode

Analysis / Requirements

Requirement	Priority
R1.1	2
R1.2	1
R1.3	3
R2.1	1
R2.2	2
R3	1

- Requirements Prioritization
    - 1-3 scale (most - least critical to project success)
- Business Requirements



R1. The application shall send an SMS text containing the joke of the day.

- R1.1. SMS texts shall be sent at the same time every day.
- R1.2. The SMS text shall contain a message with the joke of the day.
- R1.3. The SMS text, shall inform the user that they can refer to the app to view all sent jokes.

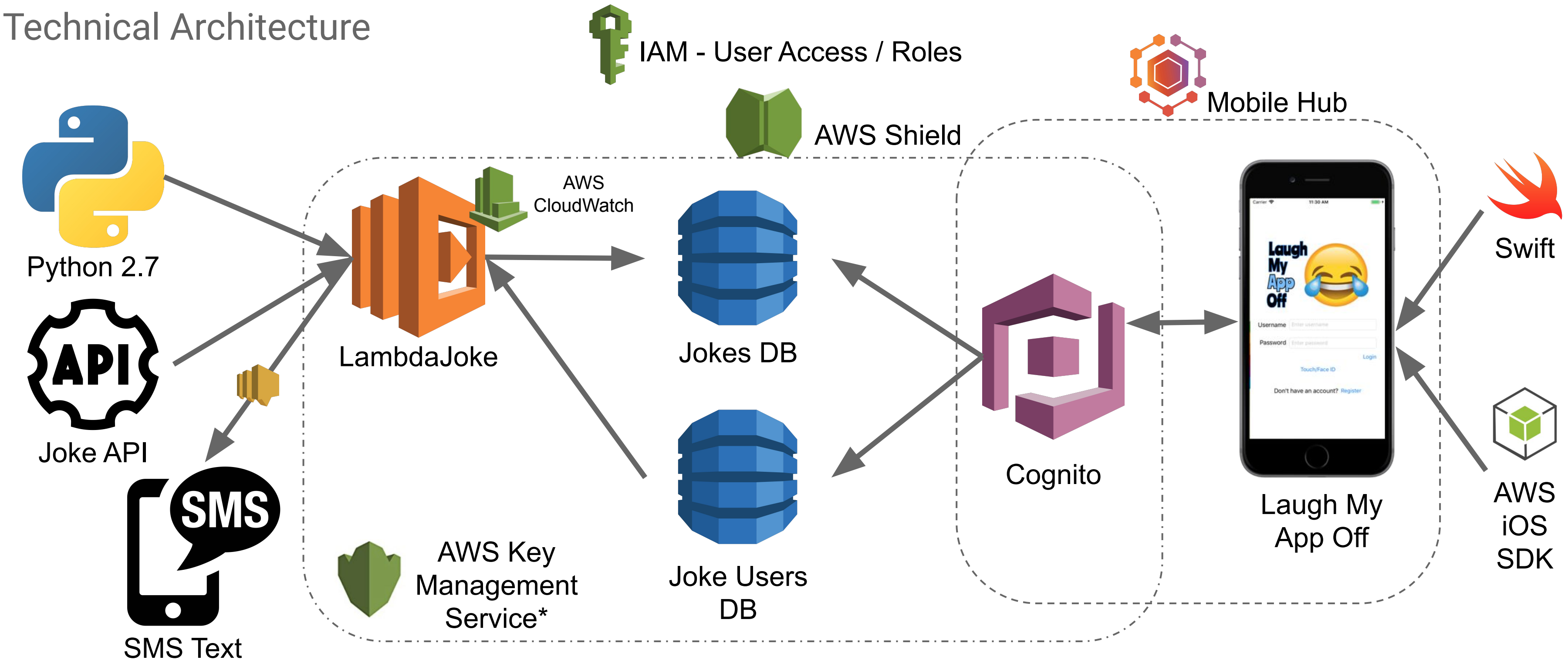
R2. The application shall allow the user to sign in or register an account

- R2.1. User sign-in shall require a username and password.
    - R2.1.1 User sign-in shall allow a user to sign-in via biometrics (TouchID/FaceID) in place of username/password.
- **R2.2 User registration shall require the following information:**
    - **First name**
    - **Username**
    - **Password**
    - **Phone number**

R3. The application shall display a list of previously received jokes.

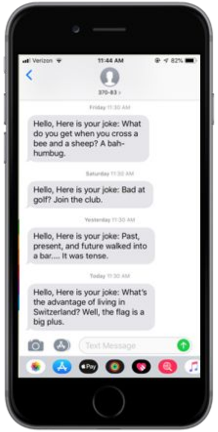
Design

Technical Architecture



Design

System Modeling



Friday 11:30 AM

Hello, Here is your joke: What do you get when you cross a bee and a sheep? A bah-humbug.

Saturday 11:30 AM

Hello, Here is your joke: Bad at golf? Join the club.

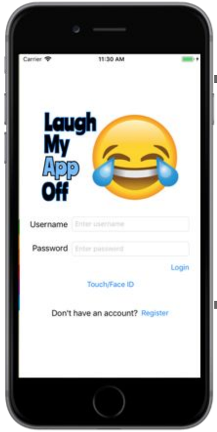
Yesterday 11:30 AM

Hello, Here is your joke: Past, present, and future walked into a bar.... It was tense.

Today 11:30 AM

Hello, Here is your joke: What's the advantage of living in Switzerland? Well, the flag is a big plus.

SMS Text



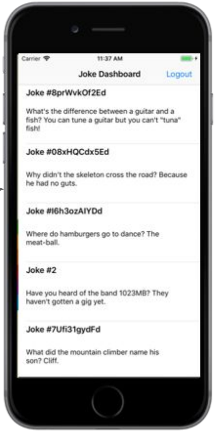
Username Enter username

Password Enter password

Don't have an account? Register

App Login Screen

Valid credentials / Biometric auth



Joke #8prWvkOf2Ed

What's the difference between a guitar and a fish? You can tune a guitar but you can't "tuna" fish!

Joke #08xHQCdx5Ed

Why didn't the skeleton cross the road? Because he had no guts.

Joke #I6h3ozAIYDd

Where do hamburgers go to dance? The meat-ball.

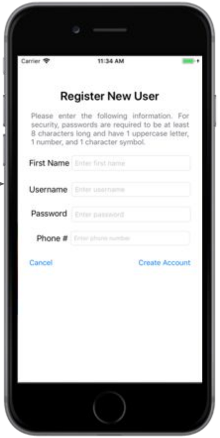
Joke #2

Have you heard of the band 1023MB? They haven't gotten a gig yet.

Joke #7Ufi31gydFd

What did the mountain climber name his son? Cliff.

Joke Dashboard



Register New User

Please enter the following information. For security, passwords are required to be at least 8 characters long and have 1 uppercase letter, 1 number, and 1 character symbol.

First Name Enter first name

Username Enter username

Password Enter password

Phone # Enter phone number

Register New User

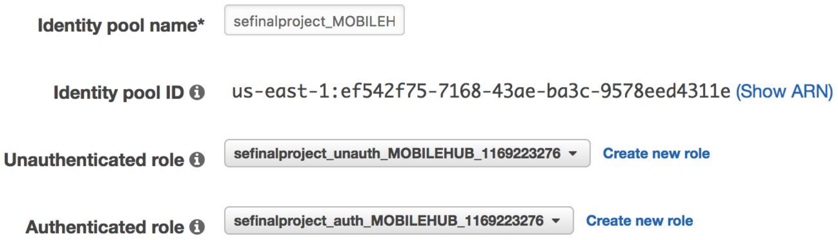
Implementation

- Method: Pilot Implementation
    - Develop app “from scratch”
- Security measures
    - iOS-side
        - Input validation
        - AWS iOS SDK
            - Set up with IAM role credentials to access AWS services from app
            - Cognito (not implemented)
    - AWS services
        - Shield
        - Cognito
        - IAM User Access / Roles
        - Access Control & Encryption in DynamoDB (not implemented)

Implementation - Cognito

Cognito Identity Pool

- Temporary AWS credentials
  - Authenticated & Unauthenticated (guests) users



Identity pool name* sefinalproject_MOBILEH

Identity pool ID ℹ us-east-1:ef542f75-7168-43ae-ba3c-9578eed4311e (Show ARN)

Unauthenticated role ℹ sefinalproject_unauth_MOBILEHUB_1169223276 ▾ Create new role

Authenticated role ℹ sefinalproject_auth_MOBILEHUB_1169223276 ▾ Create new role

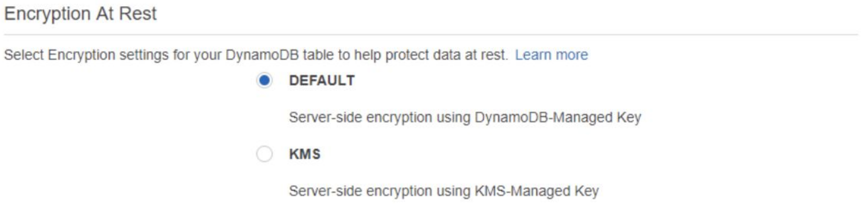
- Authentication Providers (provide tokens)
  - Cognito User Pool
  - App Client ID (from Mobile Hub)

Cognito User Pool

- User directory
- Required attributes
  - First name
  - Username
  - Password* (not stored in Cognito pool)
  - Phone number
- Password restrictions
  - 8 character minimum length
  - Must contain
    - 1 capital letter
    - 1 number
    - 1 special character
- Additional security
  - MFA
  - Attribute verification
  - Advanced security features (not included in free-tier)

Implementation – DynamoDB

- Tables
  - Joke Users
    - Username
    - First Name
    - Phone Number
  - Jokes
    - Joke ID
    - Joke



Encryption At Rest

Select Encryption settings for your DynamoDB table to help protect data at rest. Learn more

● DEFAULT

Server-side encryption using DynamoDB-Managed Key

○ KMS

Server-side encryption using KMS-Managed Key

Security Acknowledgements

- Personally Identifiable Information Perspective:
  - First Name and Phone Number mapped
  - Non sensitive
- Passwords
  - No passwords are stored in this DB
- Access Control
  - IAM Roles - determined by admin
- Encryption
  - AWS Key Management Service
    - Encryption at Rest
  - Default - key owned by DynamoDB

Implementation - Lambda

- lambda_handler
    - API Call: https://icanhazdadjoke.com/
        - Response in JSON
    - Put joke into Joke DB (DynamoDB)
        - put_item: Joke & Id
    - Query User Table (DynamoDB)
        - Scan Name & Phone Number
    - SMS Text Message
        - Traverses User Table and sends the joke to each Phone Number in the table
- Triggers
    - CloudWatch Event: Everyday at 8:05pm EST
- Execution Role: Policies
    - AWSLambdaInvocation-DynamoDB
    - AWSLambdaDynamoDBExecutionRole
    - Full Access: Lambda & DynamoDB

Security Acknowledgements

- Dependencies & Complexity
    - User and Joke Tables
        - Little complexity
        - Phone Number linked to First Name
            - No Last Name - Nonsensitive
    - Environmental Variables
        - No hardcode of actual path to the DB
        - Encryption capabilities in Key Management Service
- Internal
    - CloudTrail
        - Logs all activity in AWS account
        - Logs access to the Lambda function

Testing

OWASP Mobile Security Testing Guide (MSTG)

- "Cheat sheet" for mobile app security
  - Checklist - checks for
    - Impeding dynamic analysis and tampering
    - Device binding
    - Impeding comprehension (i.e. encryption)
  - "UnCrackables" - collection of mobile reverse engineering challenges
  - Test Case Templates

Top Mobile App Security Testing Tools 2018

- #1 OWASP ZAP (Zed Attack Proxy)
  - Malicious messages
- #2 Micro Focus
  - End-end security

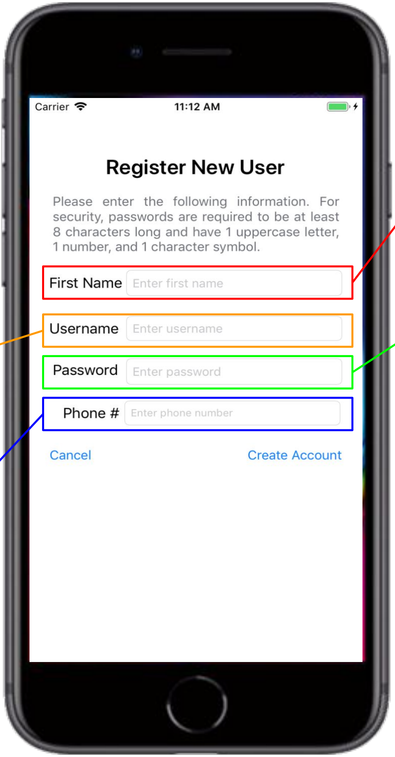
Qualys - 1Mobility

- Bring Your Own Device (BYOD) Management
- Data Loss Prevention
- Mobile Threat Management
- Compliance Enforcement

Testing - Input Validation

User Registration

- Self registration → many opportunities for error
- Input validation = good practice



Register New User

Please enter the following information. For security, passwords are required to be at least 8 characters long and have 1 uppercase letter, 1 number, and 1 character symbol.

Cannot be blank

Min. 8 characters
Must contain at least 1 number
Must contains at least 1 capital letter
Must contains at least 1 special character

Cannot be blank
Cannot be already in use

Must be numeric input (phone pad)
Must be 10 numbers long
Cannot be already in use

Most input validation for registration is covered by Cogntio, but it's always a good idea to implement as an additional measure

Maintenance

Regular maintenance and updating

- Keep up with changes to common technology
- Integrations with new tools
- Emerging vulnerabilities

When changes made

- Code reviews
    - Make sure no new vulnerabilities introduced

Good practices

- Develop maintenance & change management plans
- Rollback / disaster recovery plans

Demo

Questions?