

**MICHIGAN STATE**  
**UNIVERSITY**

# Beta Presentation

## Symantec VIP Web Framework SDK

### The Capstone Experience

Team Symantec

Ryan Casler

Jack Deters

Allen Hyunh

Gabriel Morcote

Hanlin Ye

Department of Computer Science and Engineering  
Michigan State University

Fall 2016



*From Students...*  
*...to Professionals*

# Project Overview

---

- Make integrating Symantec VIP into web apps easy
- Eliminate barrier to entry into 2FA
- Create SDKs for modern web frameworks
  - Ruby
  - Python
  - Node.js

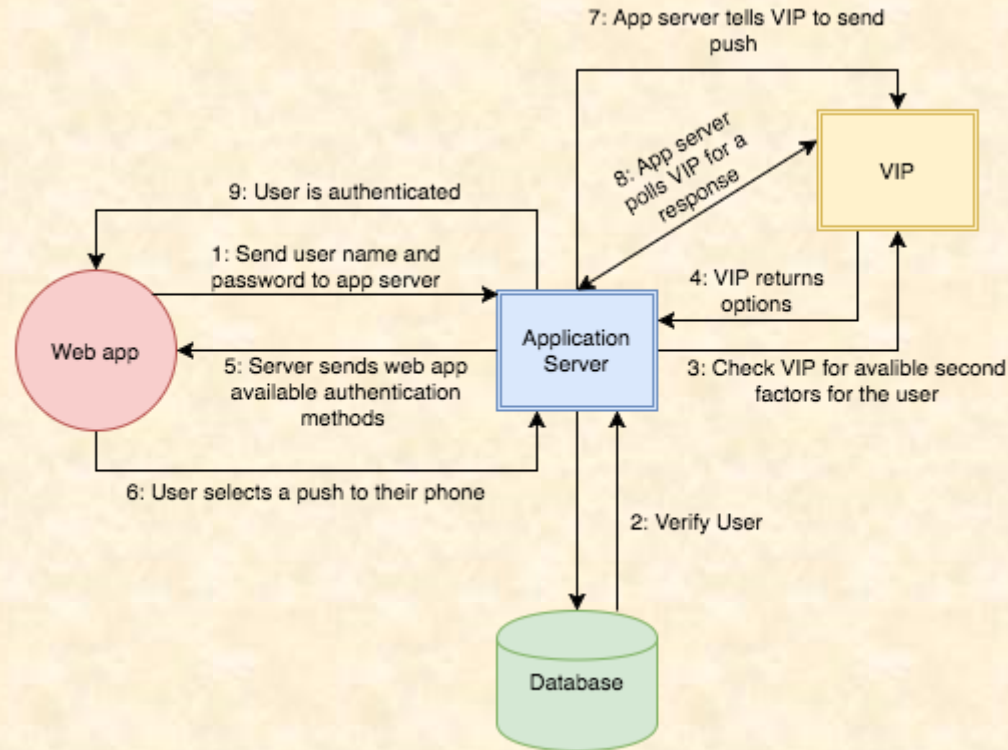


# Challenges Without Our Library

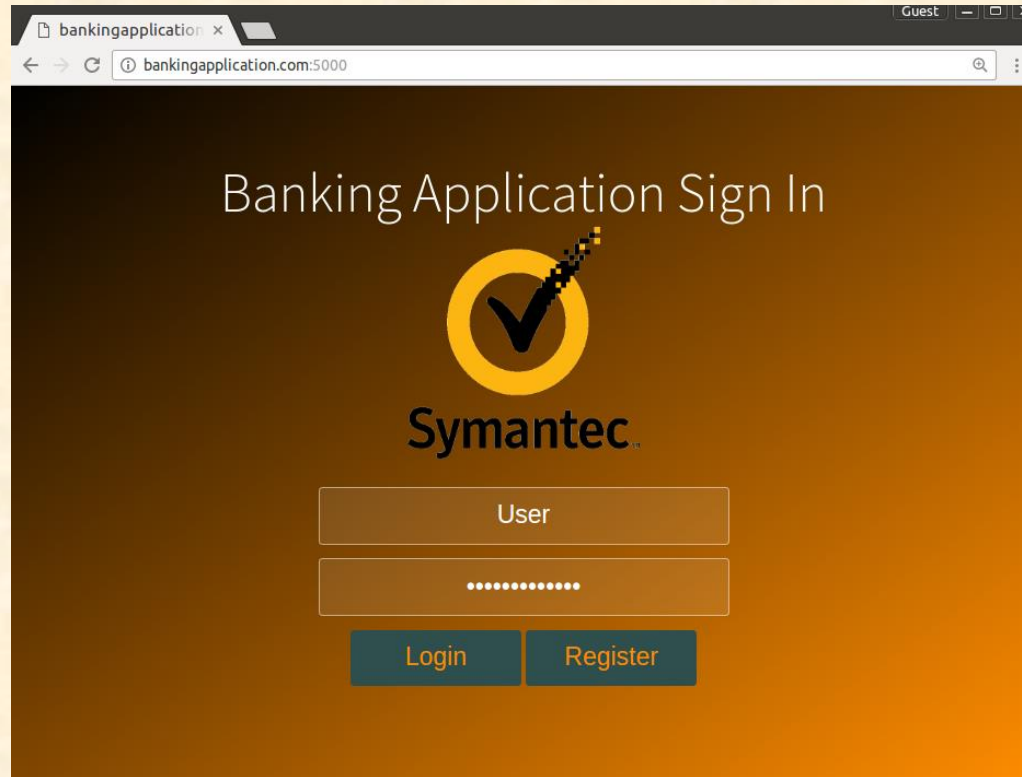
- Find a SOAP client
  - Multiple choices all with pros and cons
- Work with the client
  - Authentication
  - Deal with parameters and responses for calls
  - Corrections for errors reading the WSDL
  - Mocks for testing
- Working with VIP
  - Every call is different



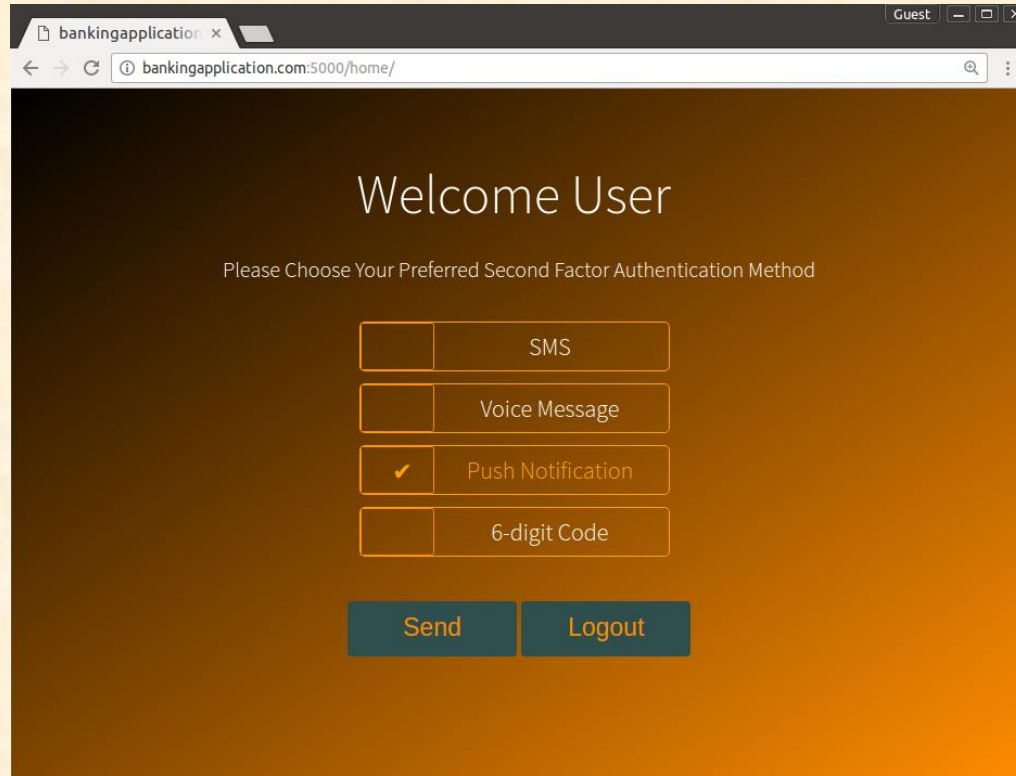
# System Architecture



# Sample App Login



# Second Factor Selection



A screenshot of a web browser window displaying a login page for a banking application. The browser's address bar shows the URL "bankingapplication.com:5000/home/". The page has a dark blue background with white text. At the top, it says "Welcome User". Below this, it prompts the user to "Please Choose Your Preferred Second Factor Authentication Method". There are four radio button options: "SMS", "Voice Message", "Push Notification" (which is selected with a checkmark), and "6-digit Code". At the bottom, there are two buttons: "Send" and "Logout".

bankingapplication x

Guest

bankingapplication.com:5000/home/

## Welcome User

Please Choose Your Preferred Second Factor Authentication Method

☐ SMS

☐ Voice Message

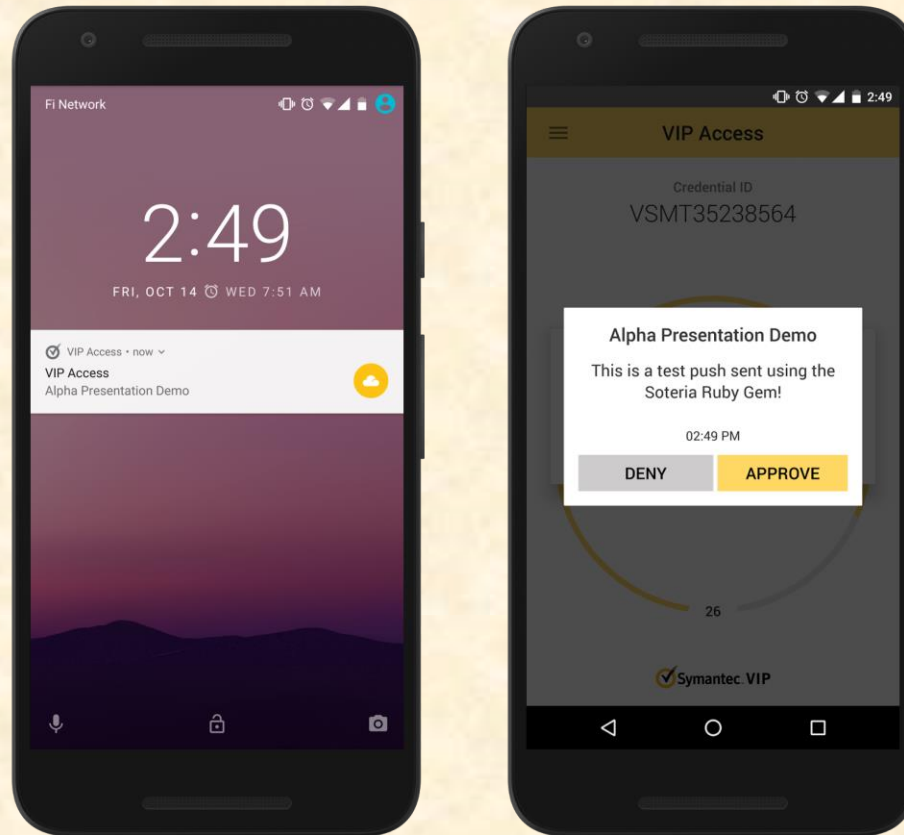
☒ Push Notification

☐ 6-digit Code

Send Logout



# Push Received By User



# Ruby Gem Documentation

The screenshot shows a web browser window displaying the Ruby Gem documentation for Soteria 0.3.0. The browser's address bar shows the URL `www.rubydoc.info/gems/soteria/0.3.0`. The page layout includes a left sidebar with a 'Class List' section containing links for 'Classes', 'Methods', and 'Files'. Below this is a search bar and a 'Top Level Namespace' section listing various classes like 'Client', 'Credential', 'CredentialTypes', 'Push', 'SMS', 'User', and 'Utilities'. The main content area is titled 'Soteria' and describes it as a gem for integrating Symantec VIP two factor authentication. It includes an 'Installation' section with instructions to add a line to the Gemfile (`gem 'soteria'`) and execute `$ bundle`, or install it manually with `$ gem install soteria`. An 'Obtaining a certificate' section explains the process of getting a certificate from the Symantec VIP manager. A right sidebar contains a 'Table of Contents' with links to '1. Installation', '2. Obtaining a certificate', '3. Usage', '4. Development', '5. Contributing', and '6. License'. The browser's user interface shows the name 'Jack' in the top right corner.

File: README — Documentatio x Jack

← → ↻ ⓘ www.rubydoc.info/gems/soteria/0.3.0 ☆ 📄 📁 📄 ⋮

**Class List**  
Classes | Methods | Files  
Search:

Top Level Namespace

- ▼ Soteria
  - Client < Object
  - Credential < Object
  - CredentialTypes < Object
  - Push < Object
  - SMS < Object
  - User < Object
  - Utilities < Object

Libraries » soteria (0.3.0) » Index » File: README

## Soteria

Soteria is a gem to integrate Symantec VIP two factor authentication into any application.

### Installation

Add this line to your application's Gemfile:

```
gem 'soteria'
```

And then execute:

```
$ bundle
```

Or install it yourself as:

```
$ gem install soteria
```

### Obtaining a certificate

To use Soteria in your project you first need a certificate from [Symantec VIP manager](#). To obtain a certificate login and go to Account -> Manage VIP Certificates -> Request a Certificate. From there follow the directions to create a new certificate. On the download screen select the PKCS#12 format and enter the password you would like to use to secure the certificate.

After downloading the PKCS#12 certificate, you must split it into a public and private key. To do so run the following two commands.

Extract the private key:

**Table of Contents**

1. Installation
2. Obtaining a certificate
3. Usage
4. Development
5. Contributing
6. License





# What's left to do?

---

- Clean up code base
- Publish sample apps and libraries
- Finish documentation



# Questions?

---

?

?

?

?

?

?

?

?

?

