Cactusforce 2023

Daniel Peter

Salesforce Practice Lead, Robots & Pencils
@danieljpeter

Making Your Slack + Salesforce Apps Even Cooler with Salesforce Functions

Thank You Sponsors!

Saguaro

validity

EzProtect By Adaptus

Agave

FormAssembly

REVENUEGRID

fileviewer

provar Test Automation for Salesforce®

{/code.scan}® by autorabit

Prickly Pear

Prodly™

Getforce

PHAROS

apps associates®

Panaya

MAMBO MERGE

Sweep

OPSERA

INKIT

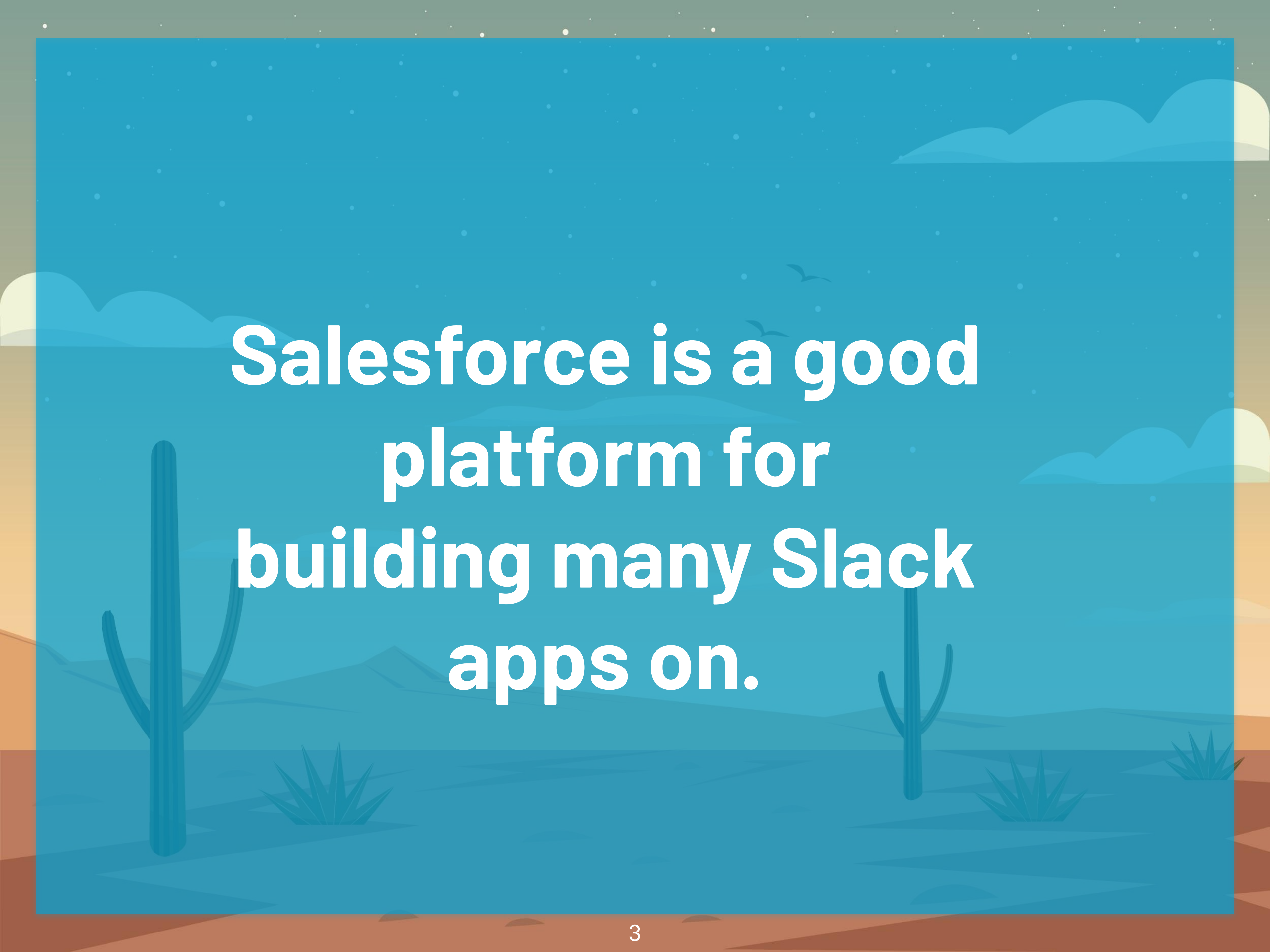
VARONIS

valence NATIVE SALESFORCE INTEGRATION

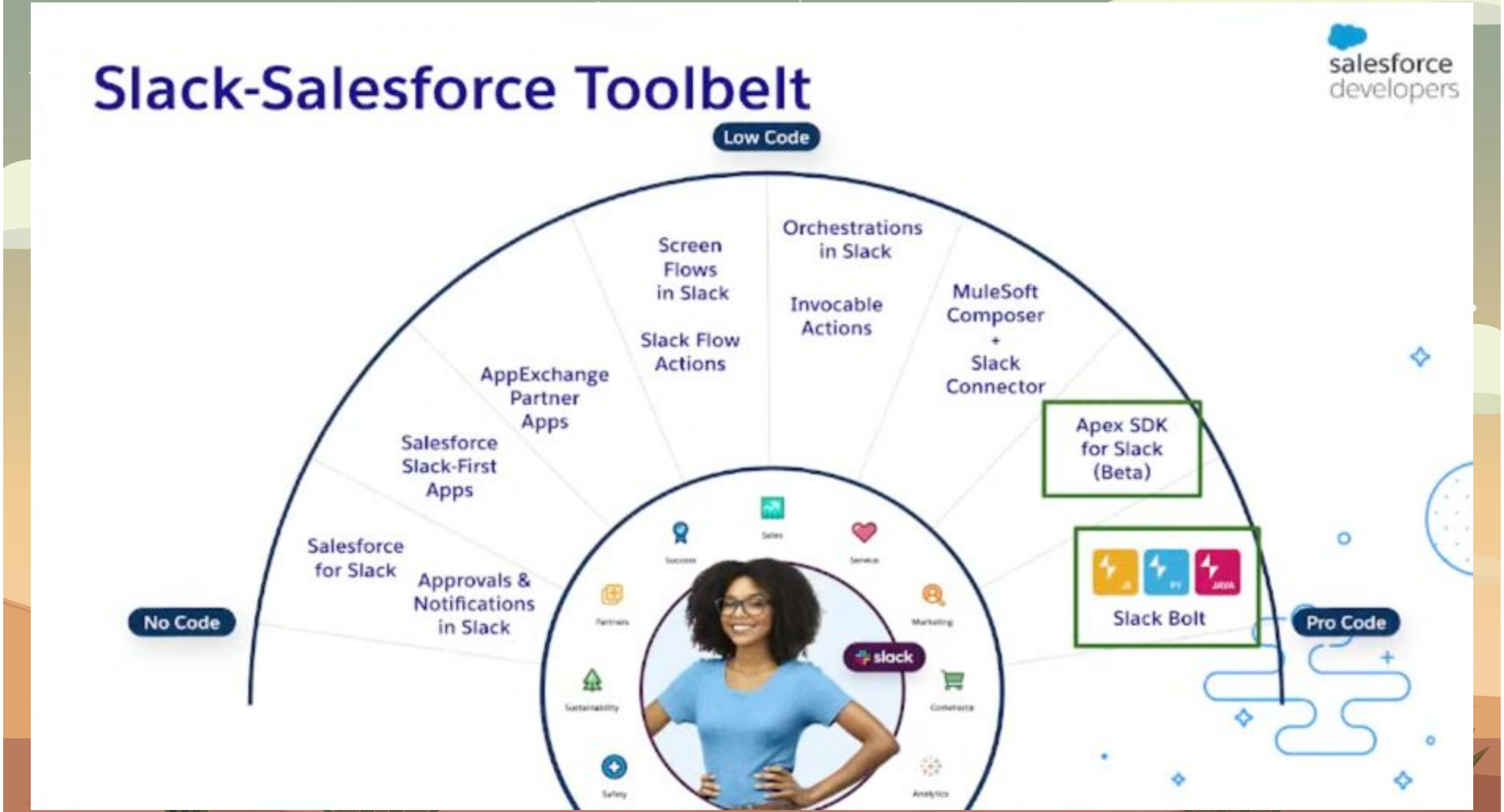
À la carte

salesforce architects

MH2X



Salesforce is a good platform for building many Slack apps on.



Slack-Salesforce Toolbelt

salesforce developers

Low Code

No Code

Salesforce for Slack

Approvals & Notifications in Slack

Salesforce Slack-First Apps

AppExchange Partner Apps

Screen Flows in Slack

Slack Flow Actions

Orchestrations in Slack

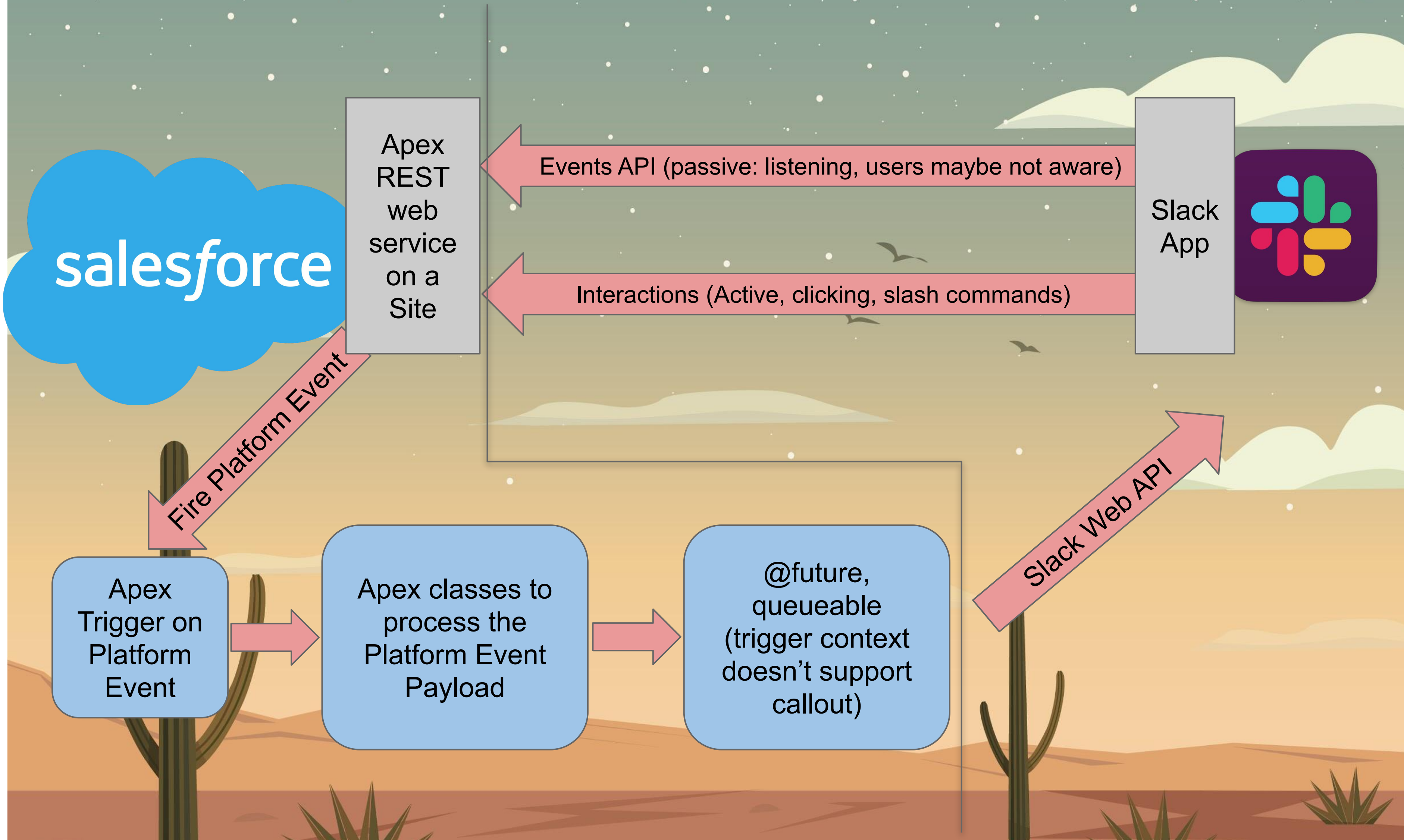
Invocable Actions

MuleSoft Composer + Slack Connector

Apex SDK for Slack (Beta)

Slack Bolt

Pro Code



salesforce

Apex REST web service on a Site

Slack App

Events API (passive: listening, users maybe not aware)

Interactions (Active, clicking, slash commands)

Fire Platform Event

Apex Trigger on Platform Event

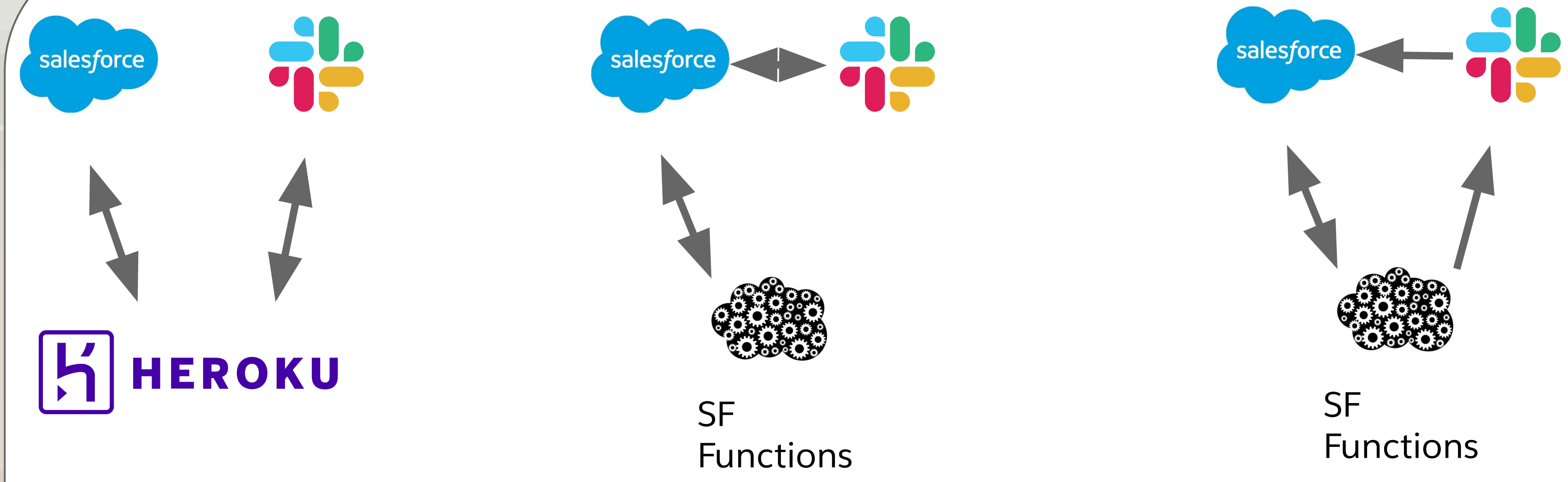
Apex classes to process the Platform Event Payload

@future, queueable (trigger context doesn't support callout)

Slack Web API

Slack/Salesforce Integrations

Many options, different tradeoffs



SF Functions

SF Functions

Heroku middleware

Salesforce middleware, Functions compute

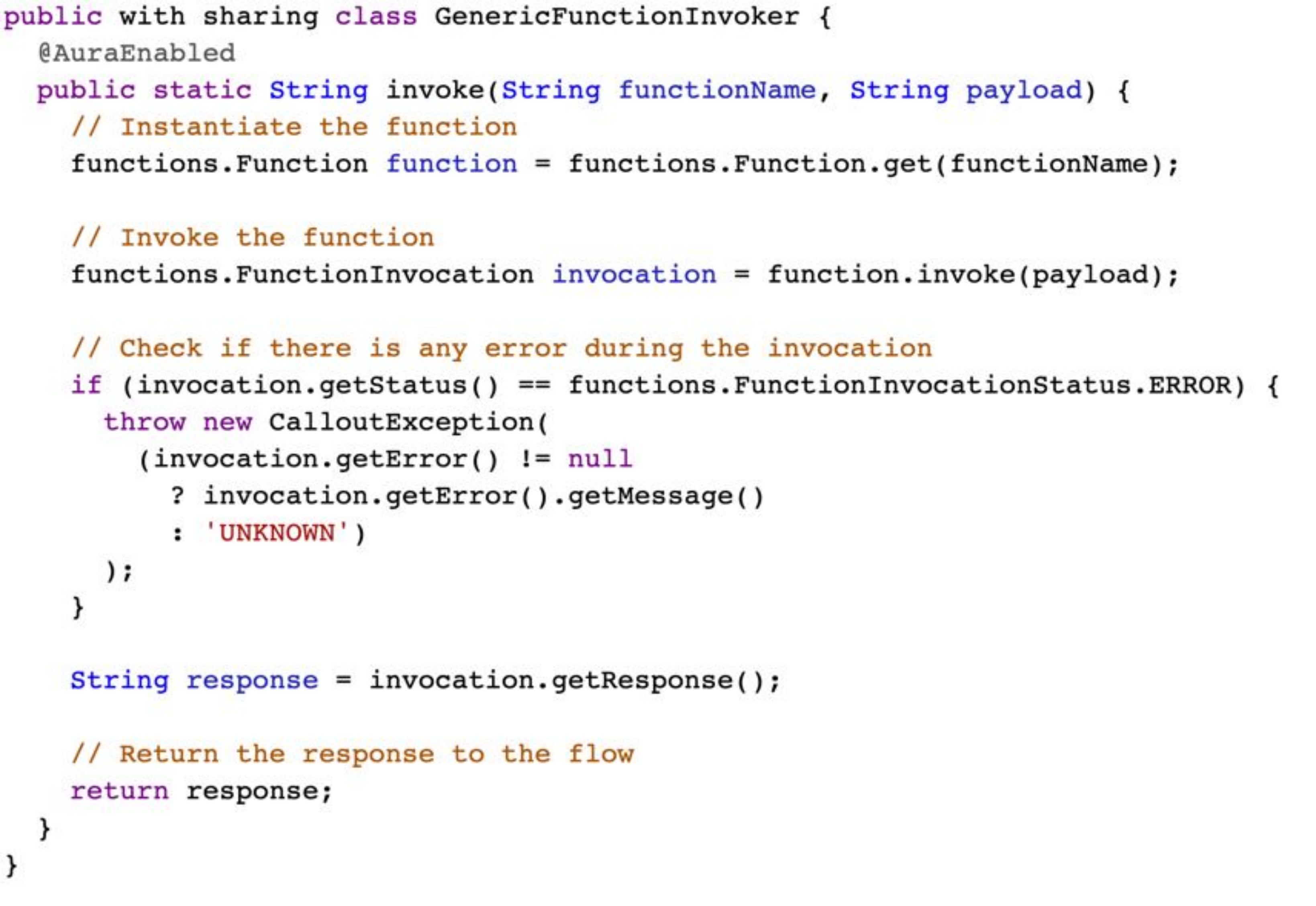
Salesforce listener, Functions compute / middleware

Salesforce as a Slack Platform

Pure Apex vs. Apex SDK

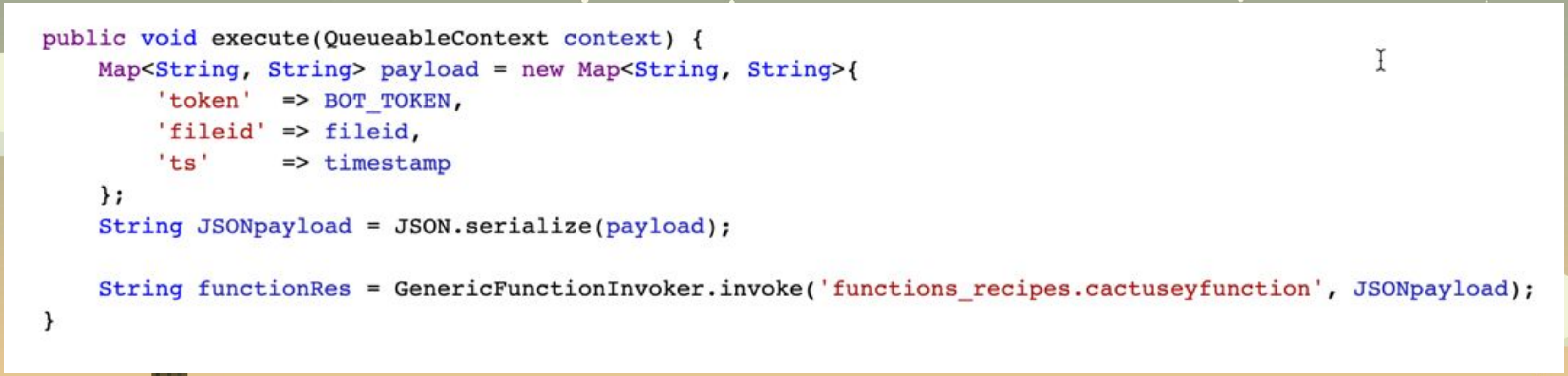
- Pure Apex is a lot of extra code, but works until Apex SDK is GA.
- Pure Apex has an issue with the context. Neither sites guest user or platform events Autoprocess user can be assigned the SF Functions permission. Apex SDK solves this.
- Pure Apex is a great way to understand the how Apex SDK works for now.

Invoking the SF Function from Apex



```
public with sharing class GenericFunctionInvoker {
  @AuraEnabled
  public static String invoke(String functionName, String payload) {
    // Instantiate the function
    functions.Function function = functions.Function.get(functionName);

    // Invoke the function
    functions.FunctionInvocation invocation = function.invoke(payload);

    // Check if there is any error during the invocation
    if (invocation.getStatus() == functions.FunctionInvocationStatus.ERROR) {
      throw new CalloutException(
        (invocation.getError() != null
          ? invocation.getError().getMessage()
          : 'UNKNOWN')
      );
    }

    String response = invocation.getResponse();

    // Return the response to the flow
    return response;
  }
}
```

Invoking the SF Function from Apex



```
public void execute(QueueableContext context) {
    Map<String, String> payload = new Map<String, String>{
        'token'  => BOT_TOKEN,
        'fileid' => fileid,
        'ts'     => timestamp
    };
    String JSONpayload = JSON.serialize(payload);

    String functionRes = GenericFunctionInvoker.invoke('functions_recipes.cactuseyfunction', JSONpayload);
}
```

Show stopper!

Neither Guest Site User or Autoproc user can execute a function!

Escalation hack.

Escalation Hack

Neither Guest Site User or Autoproc user can execute a function!

Escalation hack.

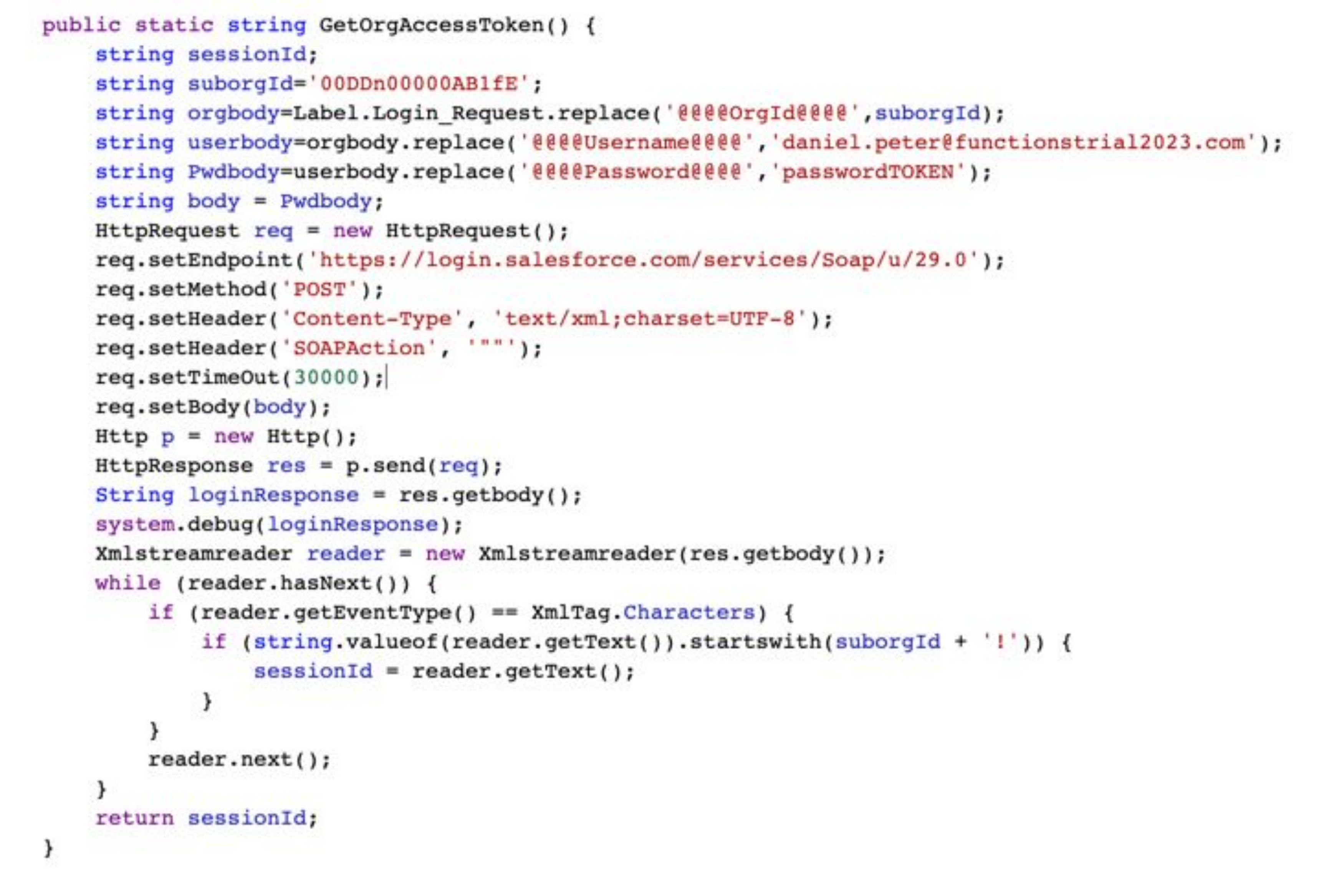
Escalation Hack

Autoproc user posts

- Org Id
- Username
- pw

to legacy soap login endpoint:

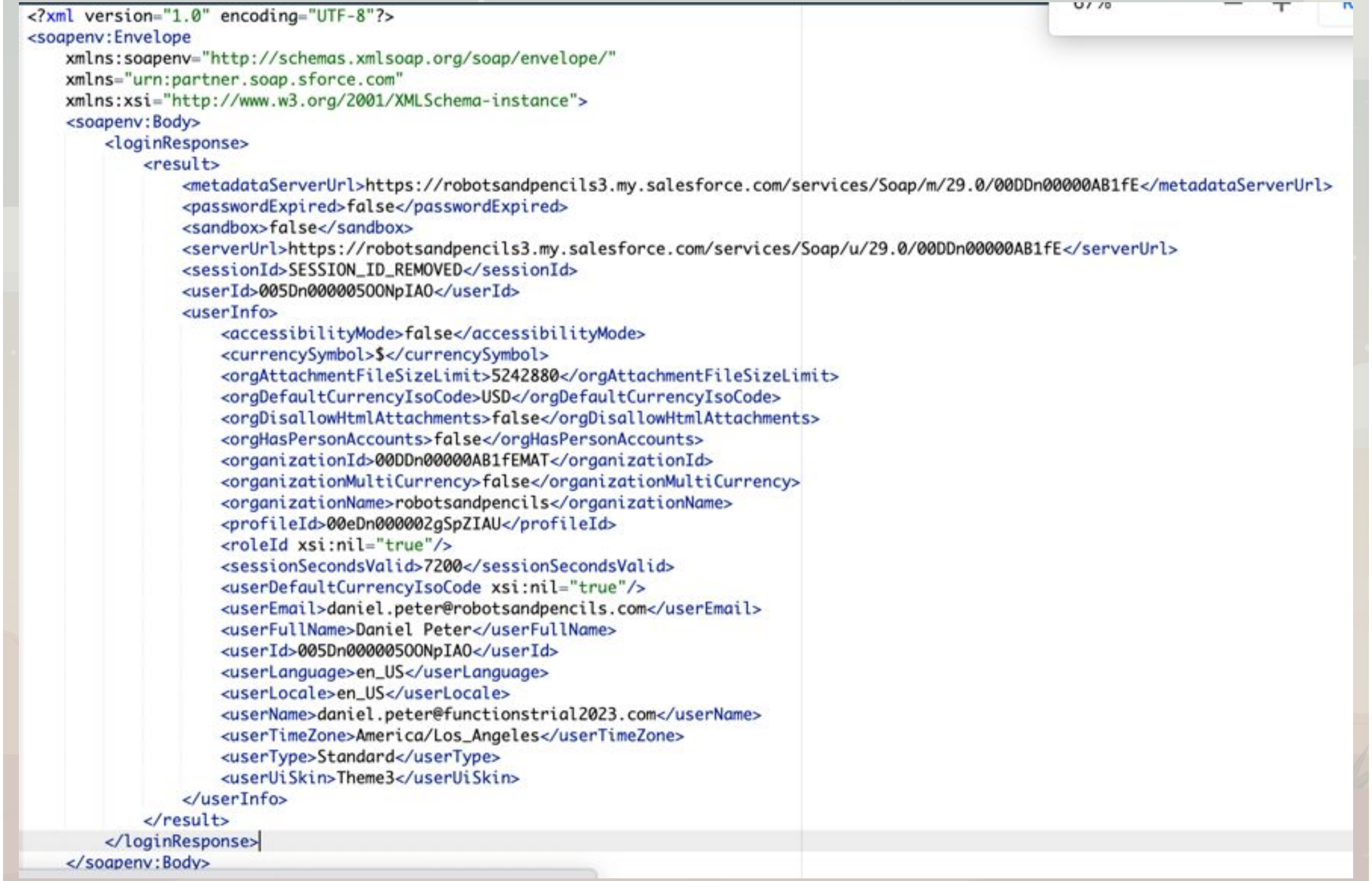
```
https://login.salesforce.com/services/Soap/u/29.0
```



```
public static string GetOrgAccessToken() {
    string sessionId;
    string suborgId='00DDn00000AB1fE';
    string orgbody=Label.Login_Request.replace('@@@@OrgId@@@@',suborgId);
    string userbody=orgbody.replace('@@@@Username@@@@','daniel.peter@functionstrial2023.com');
    string Pwdbody=userbody.replace('@@@@Password@@@@','passwordTOKEN');
    string body = Pwdbody;
    HttpRequest req = new HttpRequest();
    req.setEndpoint('https://login.salesforce.com/services/Soap/u/29.0');
    req.setMethod('POST');
    req.setHeader('Content-Type', 'text/xml;charset=UTF-8');
    req.setHeader('SOAPAction', '""');
    req.setTimeOut(30000);
    req.setBody(body);
    Http p = new Http();
    HttpResponse res = p.send(req);
    String loginResponse = res.getbody();
    system.debug(loginResponse);
    Xmlstreamreader reader = new Xmlstreamreader(res.getbody());
    while (reader.hasNext()) {
        if (reader.getEventType() == XmlTag.Characters) {
            if (string.valueof(reader.getText()).startswith(suborgId + '!')) {
                sessionId = reader.getText();
            }
        }
        reader.next();
    }
    return sessionId;
}
```

Escalation Hack

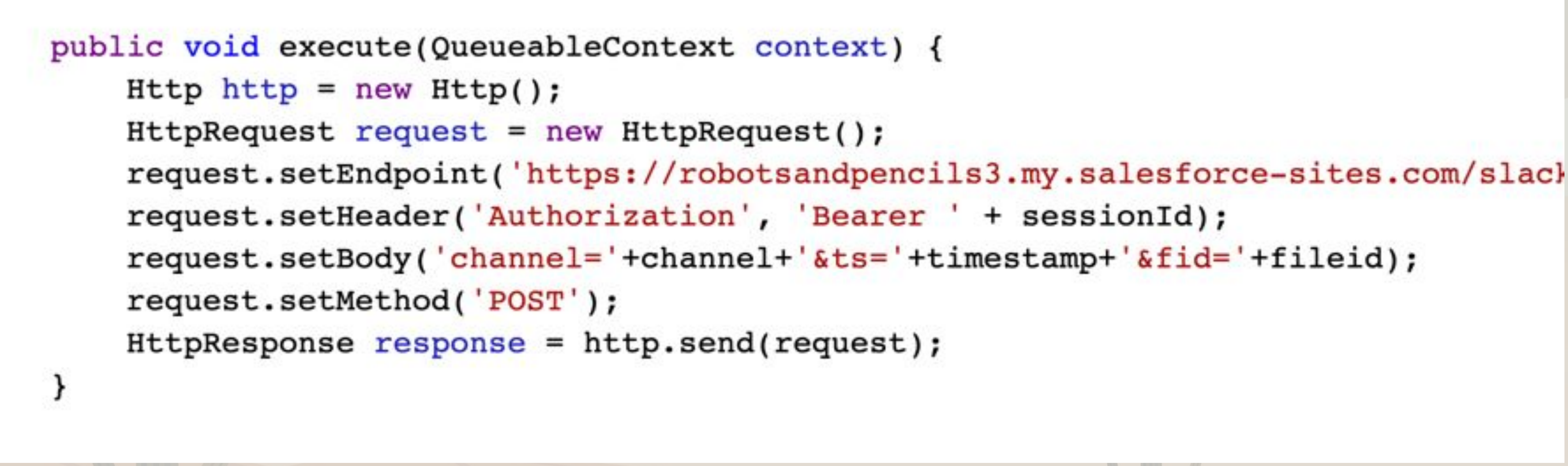
Legacy soap endpoint returns an XML response you can parse a session id out of



```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns="urn:partner.soap.sforce.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <loginResponse>
            <result>
                <metadataServerUrl>https://robotsandpencils3.my.salesforce.com/services/Soap/m/29.0/00DDn00000AB1fE</metadataServerUrl>
                <passwordExpired>false</passwordExpired>
                <sandbox>false</sandbox>
                <serverUrl>https://robotsandpencils3.my.salesforce.com/services/Soap/u/29.0/00DDn00000AB1fE</serverUrl>
                <sessionId>SESSION_ID_REMOVED</sessionId>
                <userId>005Dn000005OONpIAO</userId>
                <userInfo>
                    <accessibilityMode>false</accessibilityMode>
                    <currencySymbol>$</currencySymbol>
                    <orgAttachmentFileSizeLimit>5242880</orgAttachmentFileSizeLimit>
                    <orgDefaultCurrencyIsoCode>USD</orgDefaultCurrencyIsoCode>
                    <orgDisallowHtmlAttachments>false</orgDisallowHtmlAttachments>
                    <orgHasPersonAccounts>false</orgHasPersonAccounts>
                    <organizationId>00DDn00000AB1fEMAT</organizationId>
                    <organizationMultiCurrency>false</organizationMultiCurrency>
                    <organizationName>robotsandpencils</organizationName>
                    <profileId>00eDn000002gSpZIAU</profileId>
                    <roleId xsi:nil="true"/>
                    <sessionSecondsValid>7200</sessionSecondsValid>
                    <userDefaultCurrencyIsoCode xsi:nil="true"/>
                    <userEmail>daniel.peter@robotsandpencils.com</userEmail>
                    <userFullName>Daniel Peter</userFullName>
                    <userId>005Dn000005OONpIAO</userId>
                    <userLanguage>en_US</userLanguage>
                    <userLocale>en_US</userLocale>
                    <userName>daniel.peter@functionstrial2023.com</userName>
                    <userTimeZone>America/Los_Angeles</userTimeZone>
                    <userType>Standard</userType>
                    <userUiSkin>Theme3</userUiSkin>
                </userInfo>
            </result>
        </loginResponse>
    </soapenv:Body>
```

Escalation Hack

With that session Id you can call a custom apex REST webservice and change the user context.



```
public void execute(QueueableContext context) {
    Http http = new Http();
    HttpRequest request = new HttpRequest();
    request.setEndpoint('https://robotsandpencils3.my.salesforce-sites.com/slack
    request.setHeader('Authorization', 'Bearer ' + sessionId);
    request.setBody('channel='+channel+'&ts='+timestamp+'&fid='+fileid);
    request.setMethod('POST');
    HttpResponse response = http.send(request);
}
```

Salesforce Functions and the Slack API

node.js FTW

- Enables the use of Slack's Node SDK
  - Makes interacting with the Slack APIs faster and easier
  - Abstracts API calls to make interaction with Slack more friendly
- Use Slack's files.upload method to upload a file to Slack.
  - Node is better suited for sending files through the Slack API, making Salesforce Functions a powerful feature for integrating Slack and Salesforce
- Use other npm packages to achieve our goal
  - We can use npm packages to turn a csv file into an image of a table, and upload that to Slack, which makes it more readable
  - We can use another npm package to get a pie chart for our census, and upload that to Slack as well

Salesforce Functions and the Slack API

node.js FTW

- https://slack.dev/bolt-js/reference
- https://github.com/muenzpraeger/salesforce-einstein-vision-apex/blob/master/src/classes/EinsteinVision_HttpBodyPart.cls

Challenges

- Permissions! Make sure to give the the Functions perm set all the CRUD/FLS it will need. It's not exactly running in the executing user's security context.
- Guest site user context - escalation hack
- Not 100% of npm packages work with SF Functions. Some work locally, but don't work remotely.
- When doing team development, make sure to have a remote origin for your repo!
- Slack trigger_id 3 second rule: message instead of modals
- Bulk DML – UOW hack
- Set up local development - deploys are slow!

Live Demo

🤞

Takeaways

- Innovate on user experience with Slack and Salesforce
- SF Functions allow for specialists to easily collaborate with what they are best at
- npm makes it easy to do hard things

Resources

- Cactusforce 2022: Slack Development for Salesforce Developers
- Apex SDK for Slack
- Salesforce Functions Recipes

thanks!
Any questions?

You can find me at:
@username
user@mail.me