

Google App Scripting

Tomáš Pitner

Tvůrčí dílna INFOKON

14. listopadu 2012

Valašské Meziříčí

Google App Scripting

- Lightweight application environment:
 - develop
 - deploy
 - run in cloud
- Runs in Google Cloud
- No third party products/tools required
- No costs

Use Cases

- Simple automation
- Administration tasks
- GCalendar manipulation
- Automated mailing
- GSite Management
- GDocs Document update
- GDocs Form processing
- Using external services

Technological Background

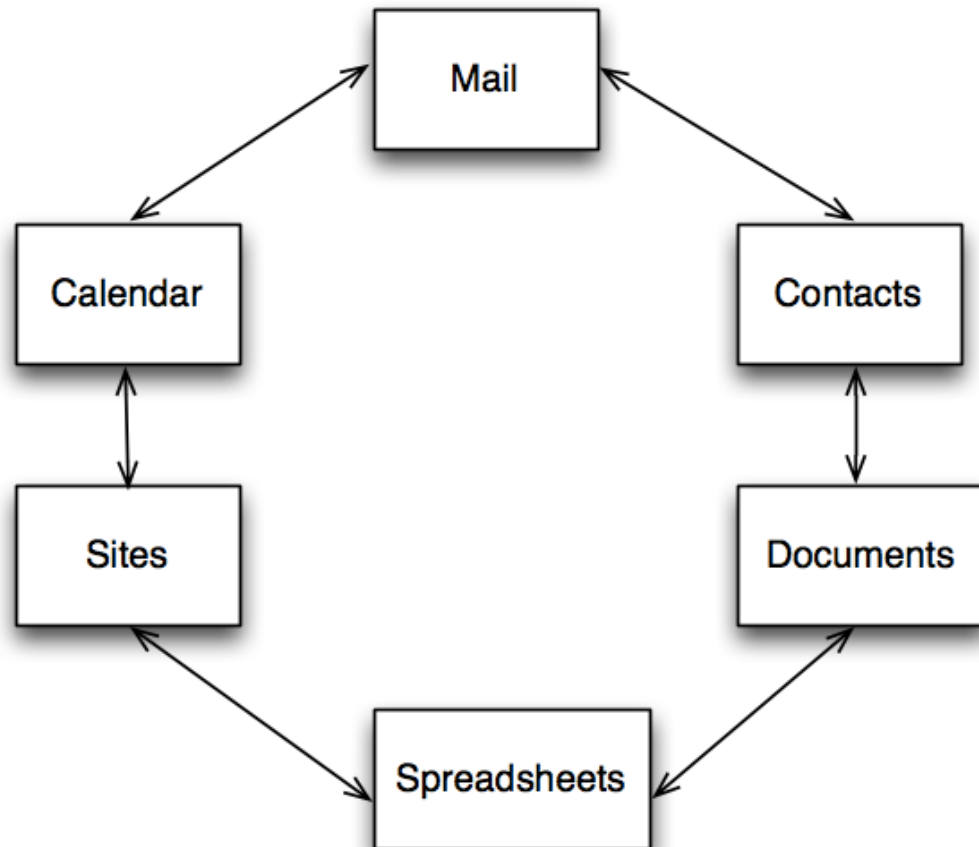
- (Server-side) JavaScript
- Many built-in objects
- Access to many Google APIs
 - Gmail
 - Maps
 - Calendar, ...
- Interact with world outside of Google
 - HTTP
 - XML
 - Mail, ...



what, **Why** and who

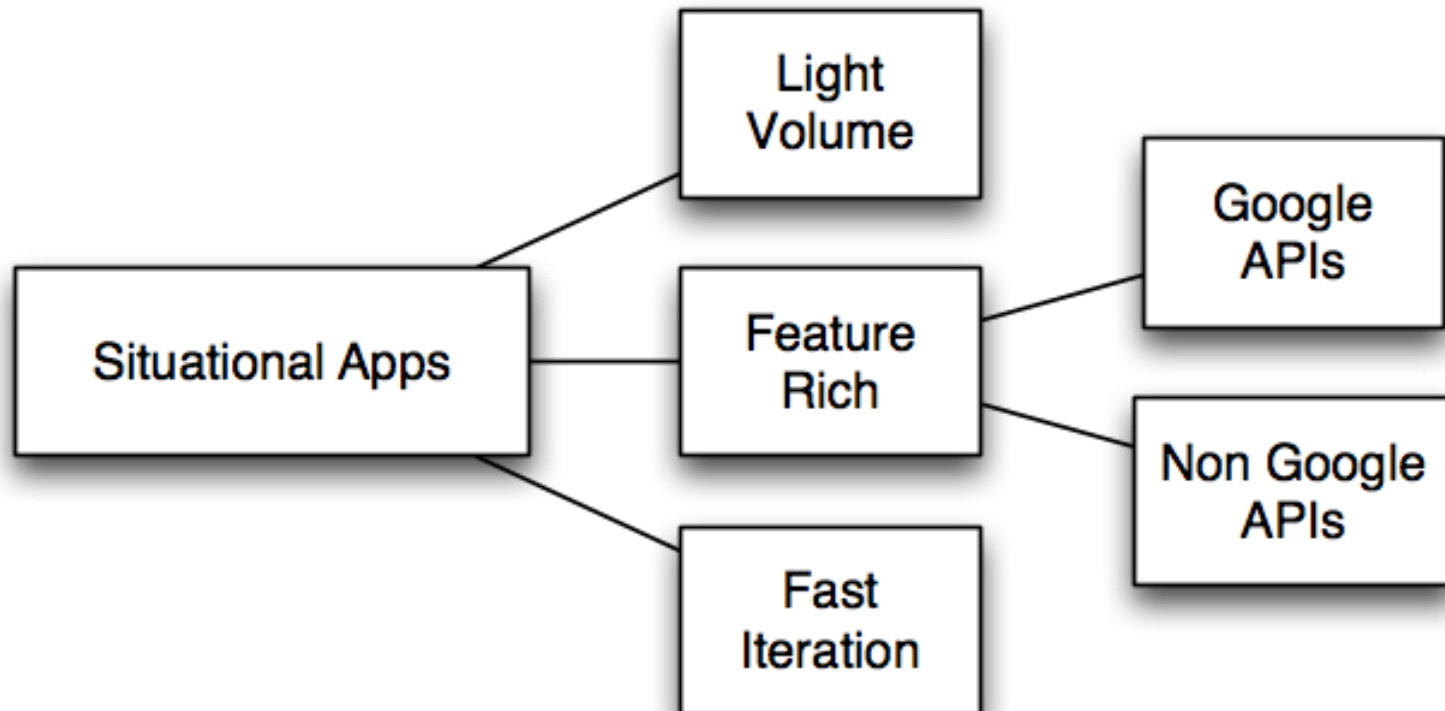
From Saurabh Gupta, Google App Script presentation for GTUG

what, **Why** and who



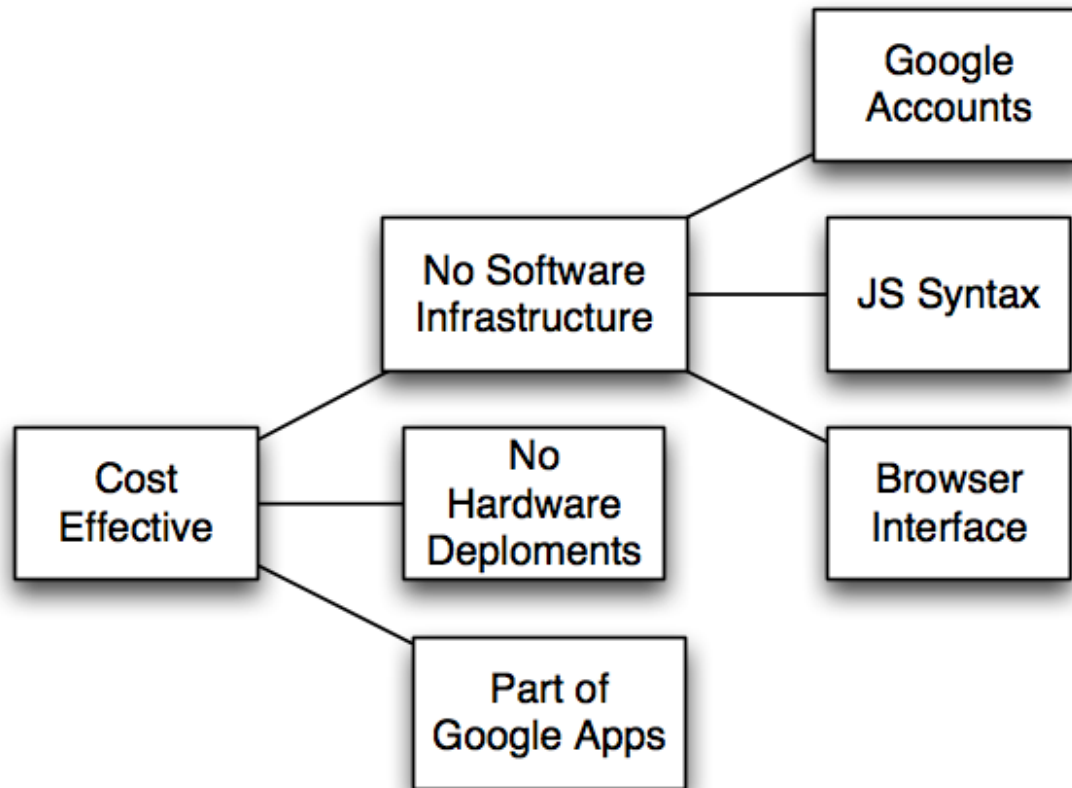
what, **Why** and who

No use case left behind



what, **Why** and who

Cost effective



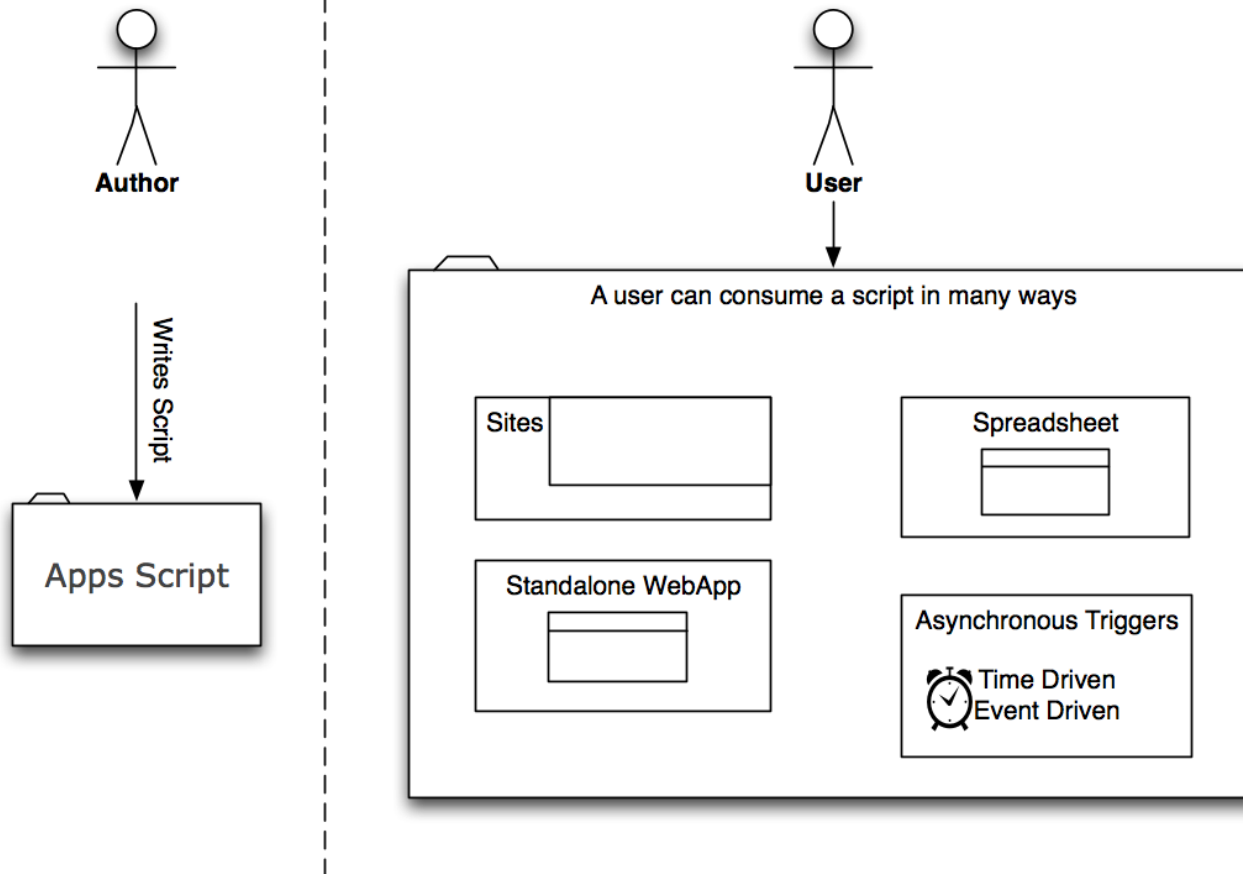
From Saurabh Gupta, Google App Script presentation for GTUG



what, why and **Who**

From Saurabh Gupta, Google App Script presentation for GTUG

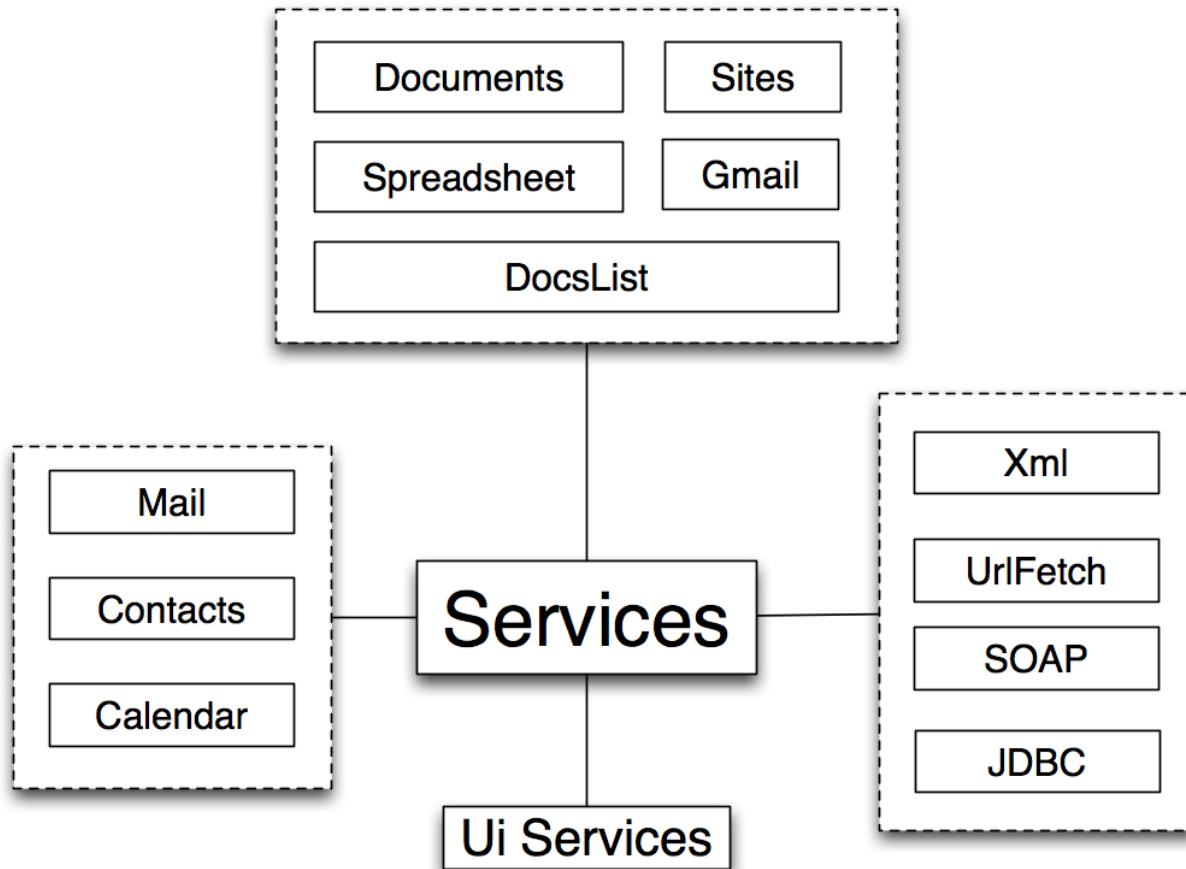
what,why and Who



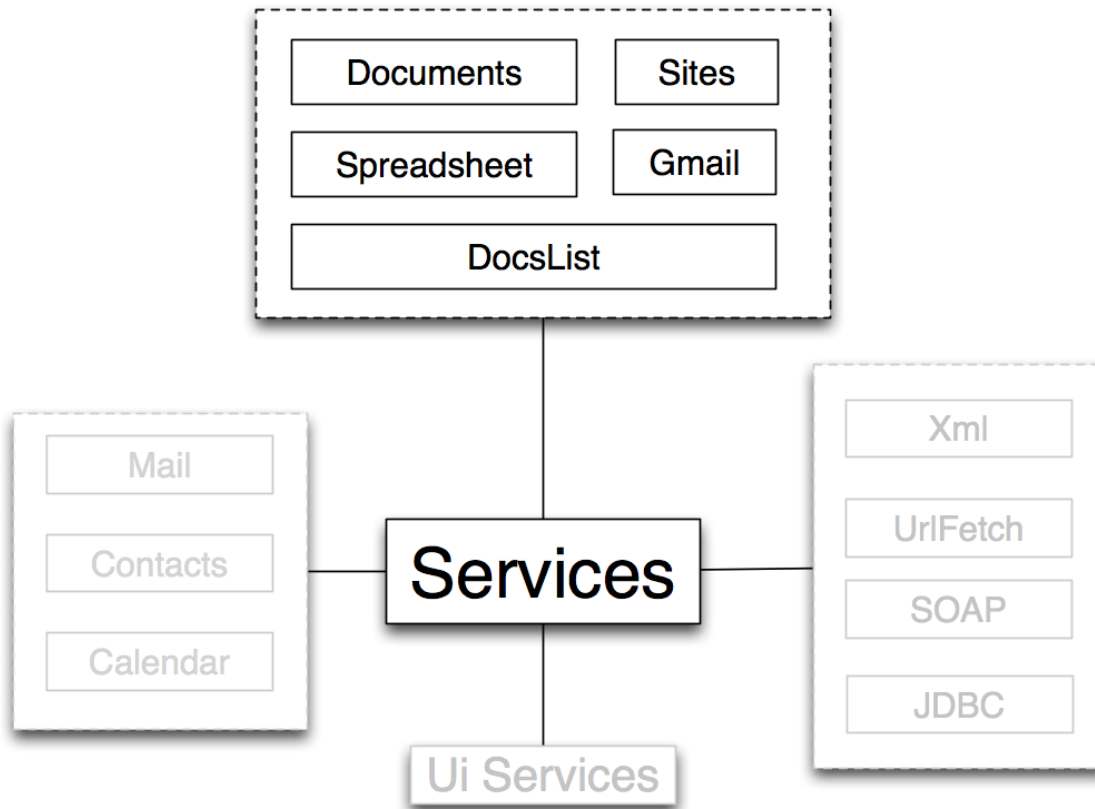
Services

- Purpose of "service" in App Scripting
- Access to particular Google service
- Mostly using built-in classes
- Or HTTP/XML-based API

Apps Script Services



Apps Script Services - Content



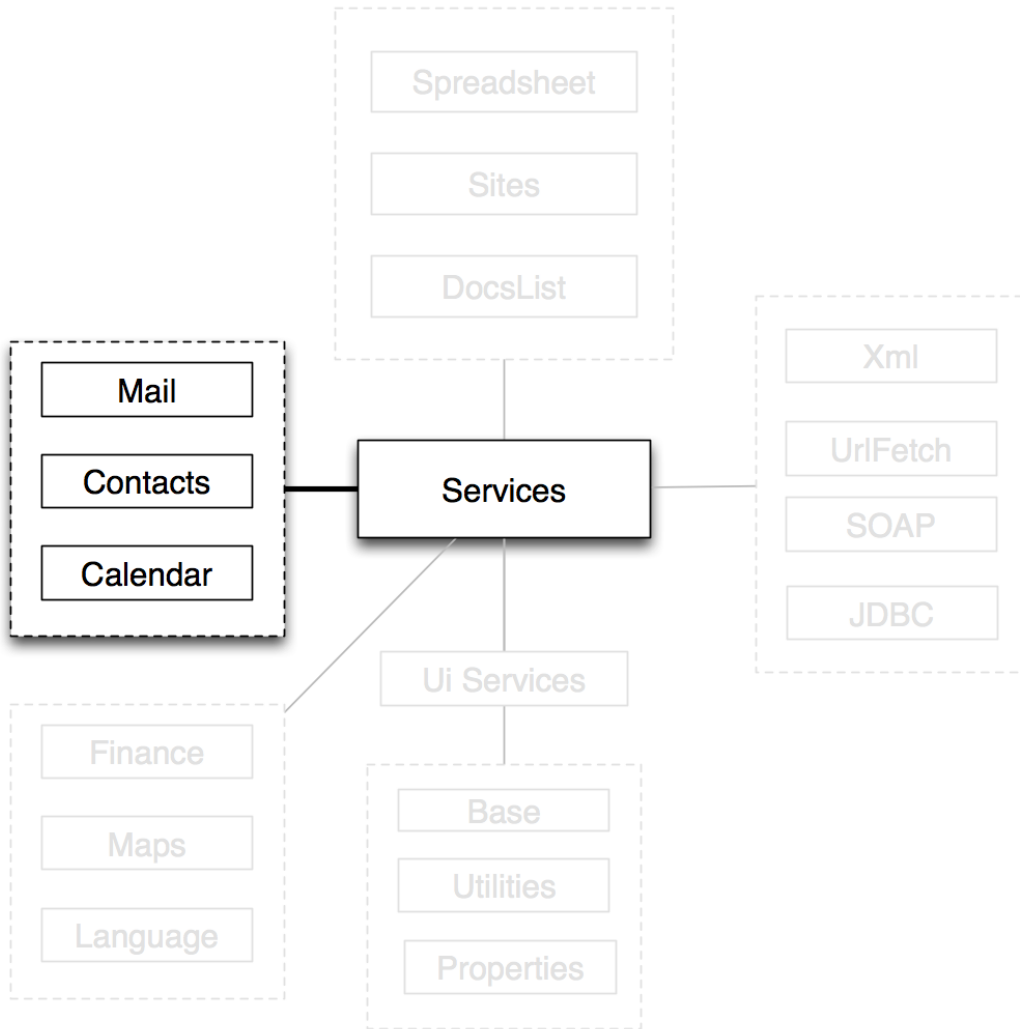
- Read and write to various Google Docs properties
- Manage Share Settings
- Automate tasks

Apps Script Services - Example

```
function map() {  
    var sheet =  
        SpreadsheetApp.getActiveSheet();  
    var cell = sheet.getRange("A2");  
    var value = cell.getValue();  
    var staticMap =  
        Maps.newStaticMap().addMarker(value);  
  
    MailApp.sendEmail(  
        "sg.appsscript@gmail.com",  
        "Map", staticMap.getMapUrl());  
}
```

From Saurabh Gupta, Google App Script presentation for GTUG

Apps Script Services - Messaging



- Services Names
 - MailApp
 - ContactsApp
 - CalendarApp

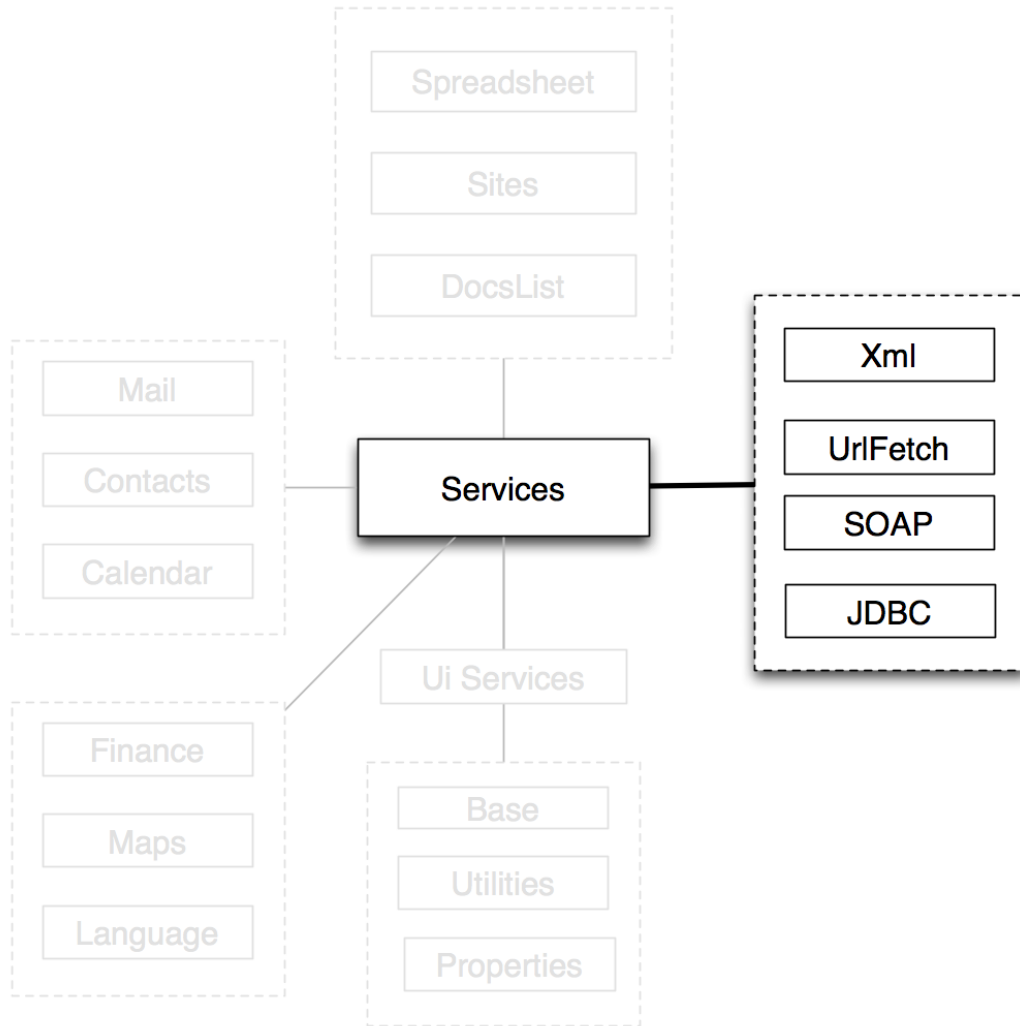
Apps Script Services - Example

```
//get calendar
var calendar =
    CalendarApp.getDefaultCalendar();

//schedule event
var startDate = new Date();
startDate.setHours(hour);
var endDate = new Date(startDate);
endDate.setHours(startDate.getHours()+2);

var event = calendar.createEvent(
    "Training:", startDate, endDate,
    {description: "Training event"});
```


Apps Script Services - 3rd Party



- Integrate other Services and 3rd party APIs
- Parse and create XML
- Fetch 3rd party data using REST and OAuth
- Connect to MySQL, Oracle, MSSQL databases

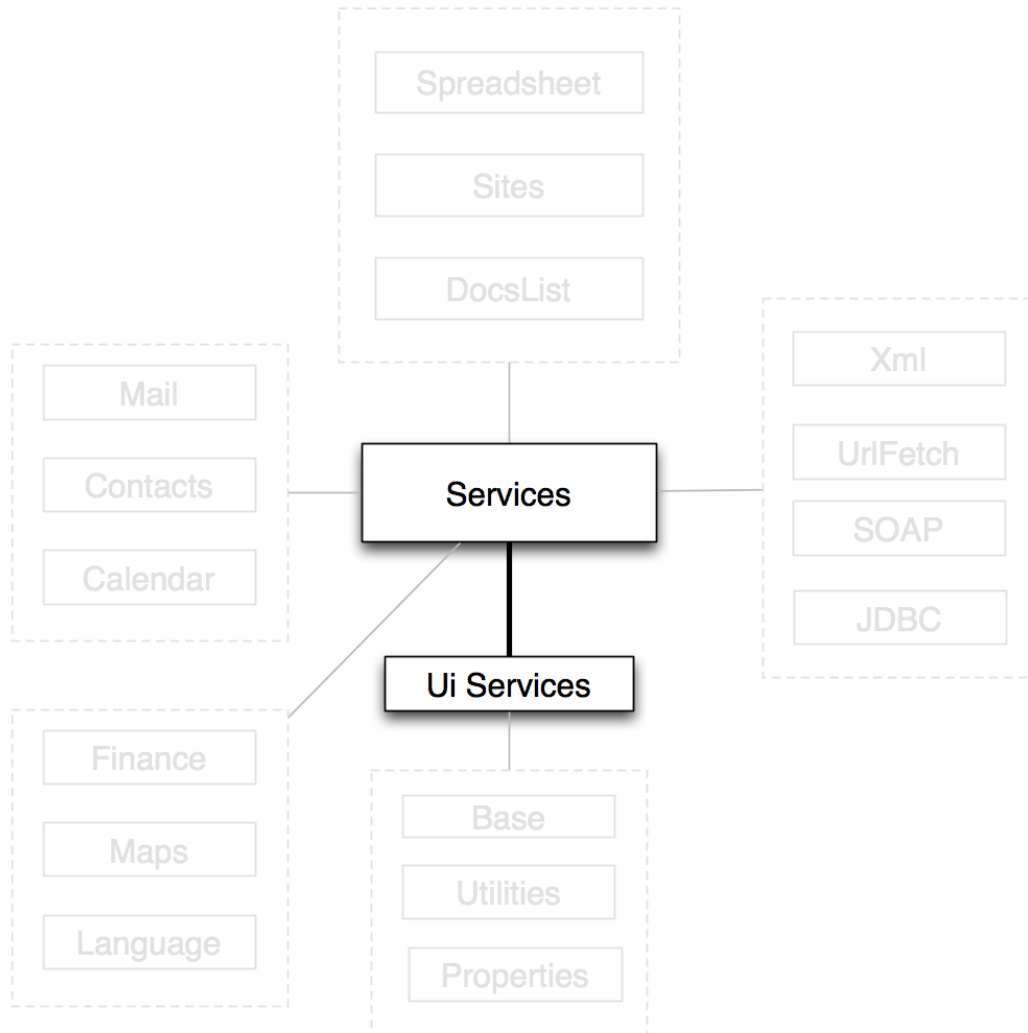
Apps Script Services - Example

```
// function to access Netflix API
var ODATA_GENRE_URL = "http://odata.netflix.com/Catalog/Genres";
function searchNetflixTitles_(
    genre, startPosition, numberOfResults) {

    var requestURL =
        ODATA_GENRE_URL+"('"+genre+"')/Titles?"
        +"$select=Name,ShortSynopsis&$format=json&$skip="
        +startPosition+"&$top="+numberOfResults;

    var content =
        UrlFetchApp.fetch(requestURL);
    var result =
        Utilities.jsonParse(
            content.getContentText());
    return result;
}
```

Apps Script Services - UI



- Underlying GWT Implementation
- A vast set of widgets Labels, Text Box, ListBox, Radio, Buttons, Forms Panels etc.
- Easily create UI Applications using these services

Apps Script Services - UI

```
// Create List Box Message Label
var listBoxMessageLabel = app.createLabel()
    .setText("1. Choose the Training Category:");
decorateLabel_(listBoxMessageLabel);
// create List Box
var categoryListBox = app.createListBox()
    .setName("categoryListBox")
    .setId("categoryListBox");
categoryListBox.addItem("Category List","0");
categoryListBox.addItem("Language Instruction","1");
categoryListBox.addItem("Computers & Electronics","2");
categoryListBox.addItem("Career & Finance","3");
categoryListBox.addItem("Healthy Living","4");
categoryListBox.addItem("Workouts","5");
decorateLabel_(categoryListBox);

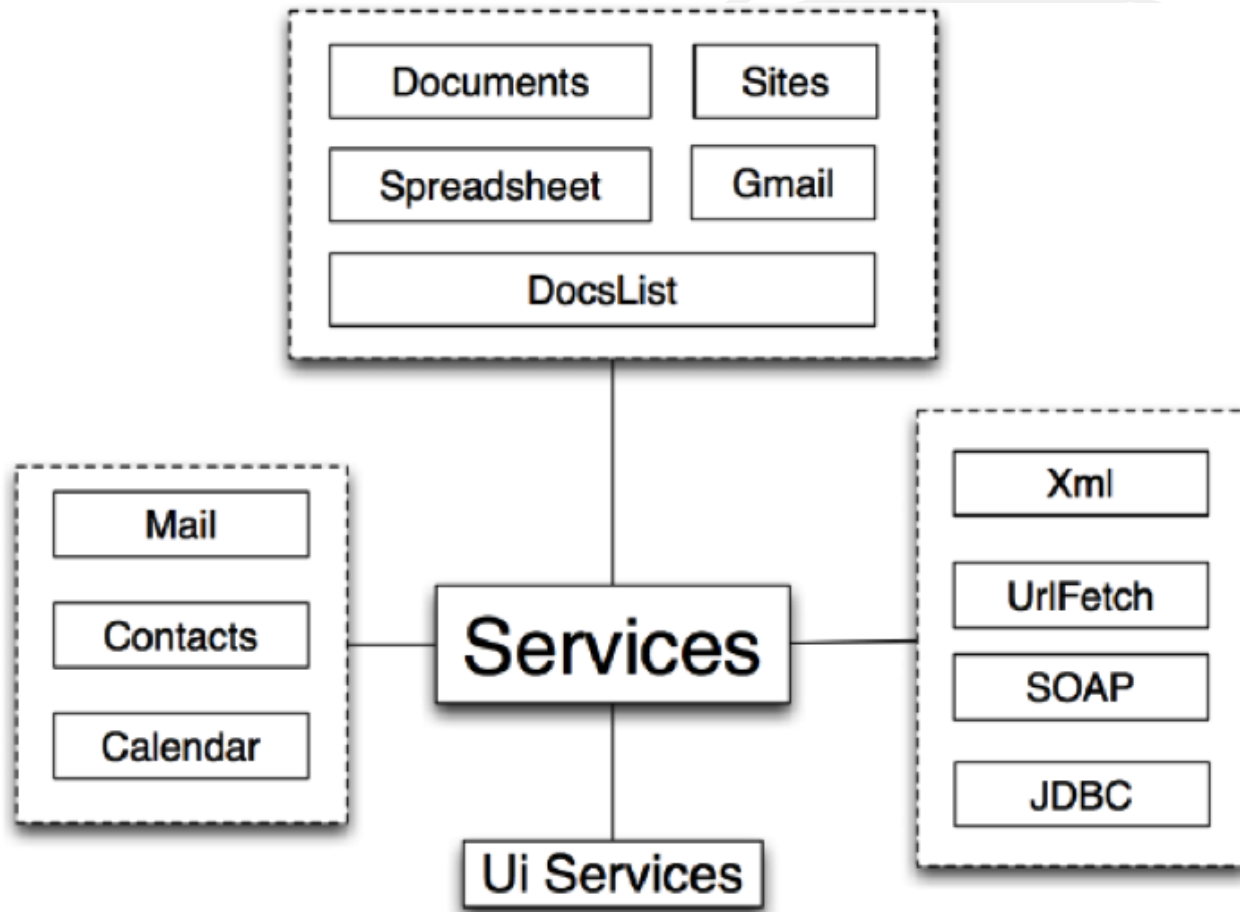
// create change handler
var categorySelectHandler =
    app.createServerChangeHandler("categorySelectionHandler_");

categorySelectHandler.addCallbackElement(mainPanel);
categoryListBox.addChangeHandler(categorySelectHandler);
```

Apps Script Services - Triggers

- Trigger = Cron's Cousin
- Triggers allow asynchronous execution of scripts
 - Helps in automation - A user no longer has to manually execute scripts
- Two types of Triggers
 - Event Driven - onEdit, onInstall, onOpen
 - Time Driven

Ecosystem:



from C. Cherubino, <http://www.slideshare.net/ccherubino/enterprise-workflow-with-apps-script>

User Interface

- Google Web Toolkit-based
 - AJAX inside but abstracted from the developer

Development Process

- Cloud-based debugger for G.App JavaScript

Limitations

- No access to corporate (internal) enterprise systems like databases
- No HTML show/edit widget in GWT
- No client-side JavaScript
- Some APIs are not mature

Resources :

Articles (Google)

<http://code.google.com/googleapps/appscript/articles.html>

Developer blog

<http://googleappscript.blogspot.com/>

Enterprise workflow samples presentation

<http://www.slideshare.net/ccherubino/enterprise-workflow-with-apps-script>

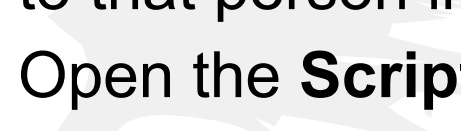
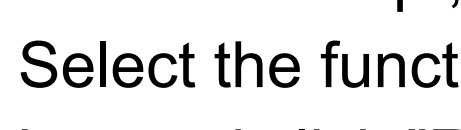
Example 1: Mailing, GDocs

Complete at http://code.google.com/googleapps/appscript/articles/sending_emails.html

Hugo Fierro, Google Apps Script Team May 2009

```
function sendEmails() {  
    var sheet = SpreadsheetApp.getActiveSheet();  
    var startRow = 2;    // First row of data to process  
    var numRows = 2;    // Number of rows to process  
    // Fetch the range of cells A2:B3  
    var dataRange = sheet.getRange(startRow, 1, numRows, 2)  
    // Fetch values for each row in the Range.  
    var data = dataRange.getValues();  
    for (i in data) {  
        var row = data[i];  
        var emailAddress = row[0];    // First column  
        var message = row[1];        // Second column  
        var subject = "Sending emails from a Spreadsheet";  
        MailApp.sendEmail(emailAddress, subject, message);  
    }  
}
```

Ex 1: Development Workflow

1. Create a new empty **Spreadsheet**
2. Prepare **data**. Every row should contain an email address in column A and the email message to be sent to that person in column B.
3. Open the **Script Editor** by clicking on the 'Tools' menu, then select 'Scripts' and 'Script editor...'.

4. **Write** the script, **Save** the Script
5. Select the function **sendEmails** in the function combo box and click "Run"

6. Check out your **email** Inbox. Messages are usually immediately delivered, but sometimes it takes a few seconds.

Real Workflows with Apps Script

- Workflow
 - Management of a sequence of steps in a business process
- Google Sites
 - Easy way to create and share web pages and sites
- Google Spreadsheet
 - Easy way to store workflow data and activity
- Google Apps Script
 - Glue that brings together to create a business process

How to Define a Workflow

- Workflow Definition
- Workflow States
 - Each State encapsulates the logic which governs the transition of the workflow
 - Defines the view that each state will create for the user
- Workflow Data
 - Data that is associated with the entire workflow including all data in the states