

mt



HI 007

007

# Offline data manipulations for everyone with Plug-Ins and SQLite

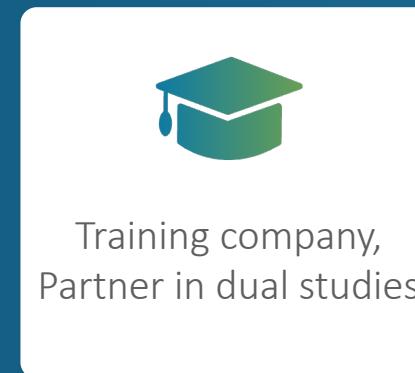
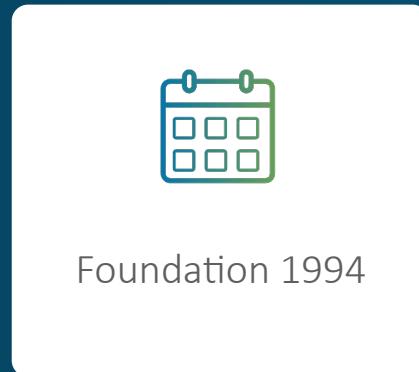
27-June-2023

Philipp Hartenfeller, Senior Consultant

KScope 2023, Aurora / Colorado

# Facts and figures.

Your partner for digital transformation.  
Individual IT solutions from one single source.



\$ whoami



## Philipp Hartenfeller

- Düsseldorf, Germany
- Master IT-Management
- Since 2016 @ MT AG GmbH
- Senior Consultant – Oracle APEX
- Mostly doing WebDev, DBs and APEX Testing (<https://lct.software>)

Blog: <https://hartenfeller.dev/blog/>



[@phartenfeller](https://twitter.com/phartenfeller)



[@phartenfeller@mastodon.social](https://phartenfeller.mastodon.social)



Oracle ACE  
Associate

## What this talk is about

- Offline APEX is no new topic, there have been a few methods already
- I am interested in **HOW** it should be usable in APEX

# What this talk is about

## How should it be accessible ?

- Low Code Interface -> No JavaScript Code wrangling
  - **APEX Plug-Ins**
- Easy to configure
  - Enter your Query and let's go!

# What this talk is about

## How should it be accessible ?    How should it work ?

- Low Code Interface -> No JavaScript Code wrangling
  - **APEX Plug-Ins**
- Easy to configure
  - Enter your Query and let's go!
- Fast
  - Handle lots of data
  - Concept to merge changes back to DB
  - Somewhat future proof

Proof of Concept: Not production ready!

# Upload sw.js

Application 120 \ Shared Components \ Static Application Files \ **Create**

Cancel

Directory

\* File Name

Content

Drag and Drop Files +  
Select or drop files here.

<input type="checkbox"/>	JS	sw.js	text/javascript	6KB	#APP_FILES#sw#MIN#.js
--------------------------	----	-------	-----------------	-----	-----------------------

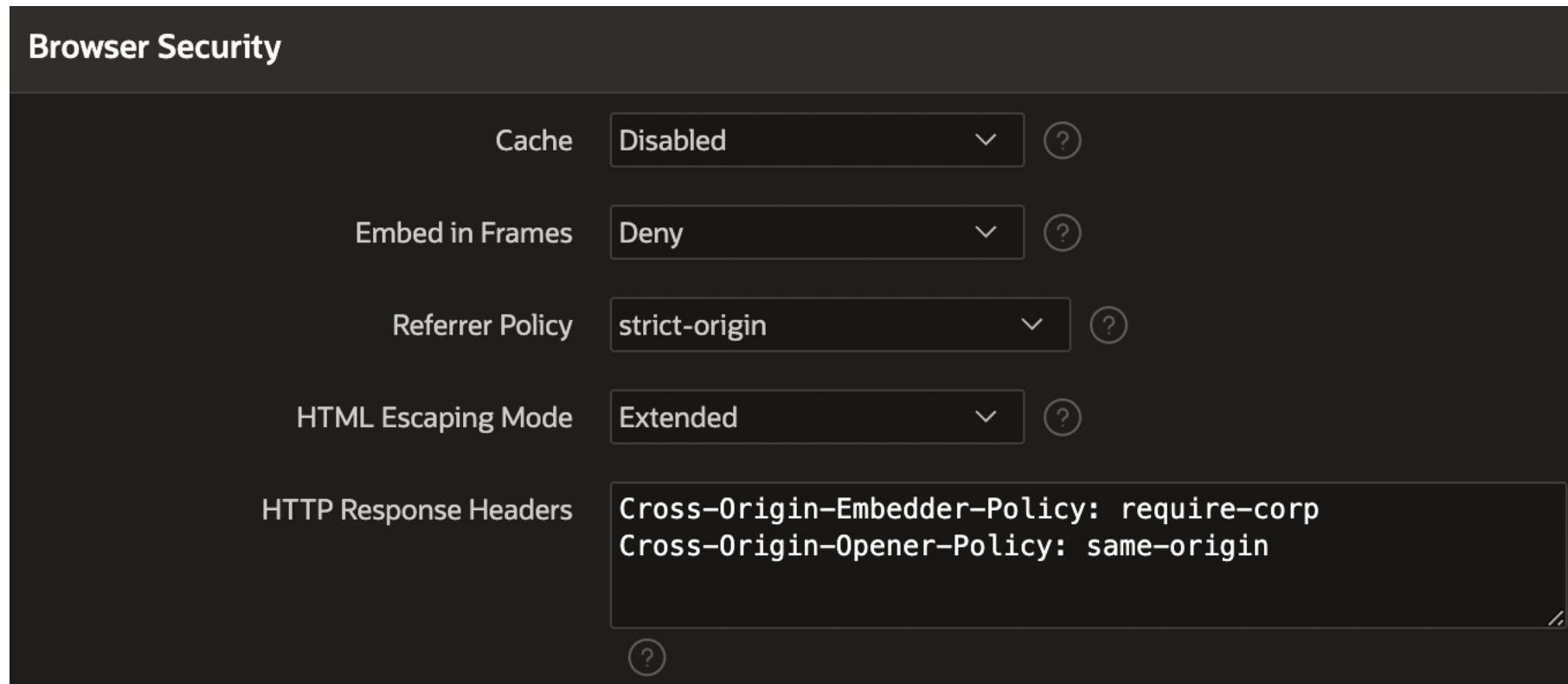
# Enable PWA features

The screenshot shows the 'Application 120' configuration screen. At the top, there's a breadcrumb navigation: 'Application 120 \ Progressive Web App'. Below the navigation, there are tabs: 'Definition', 'Security', 'Globalization', 'User Interface', and 'Progressive Web App', with 'Progressive Web App' being the active tab. A green 'Apply Changes' button is visible. Under the 'General' section, there are two toggle switches: 'Enable Progressive Web App' (which is turned on) and 'Installable' (which is also turned on). There are also tabs for 'Show All', 'General', 'Appearance', and 'Service Worker Configuration', with 'General' currently selected.

The screenshot shows the 'Service Worker Configuration' screen. At the top, there are tabs: 'Service Worker', 'Default', 'Configure Hooks', and 'File URL', with 'File URL' being the active tab. Below the tabs, there's a field labeled '\* File Reference URL' containing the value '#APP\_FILES#sw#MIN#.js'. A question mark icon is located at the bottom of this field.

Also enable  
Persistent  
Auth!

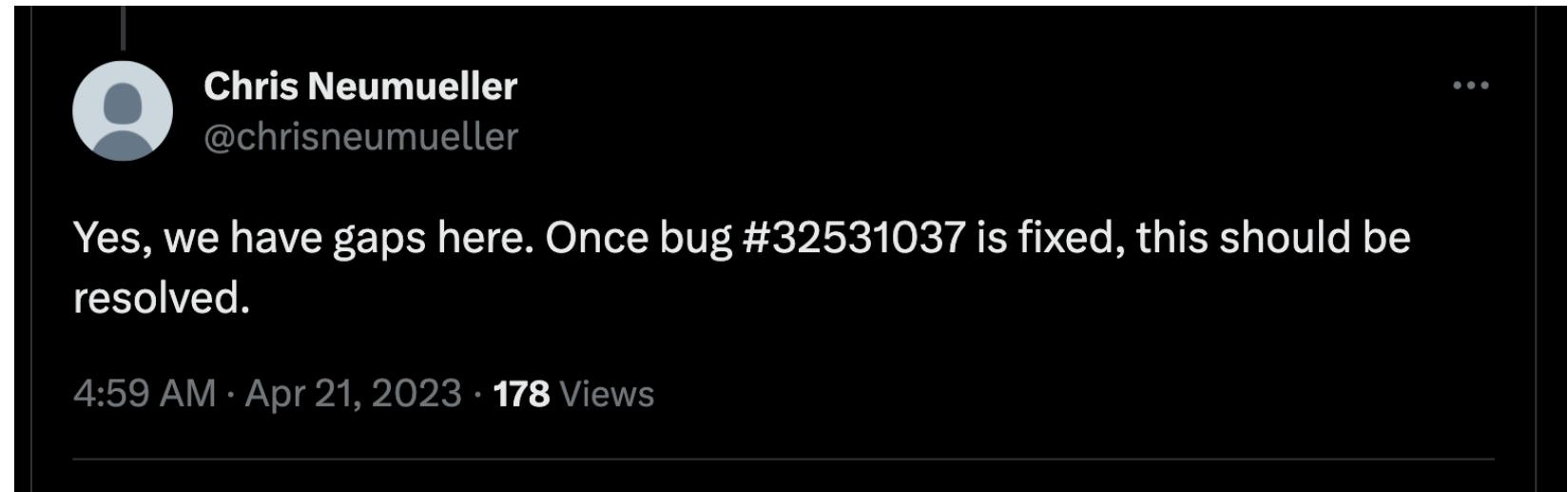
# Set required Headers



Restricts how other websites can access data of APEX pages

Currently a bug in APEX (does not apply to Plug-In files)

A [bug is filed](#)



Chris Neumueller  
@chrisneumueller

Yes, we have gaps here. Once bug #32531037 is fixed, this should be resolved.

4:59 AM · Apr 21, 2023 · 178 Views

For now, you need to set it on Proxy /  
Webserver

# Import Plug-Ins

The screenshot shows a user interface for managing plug-ins. At the top, there's a search bar with a placeholder "Select columns to search" and a "History" button. Below the search bar are buttons for "Go", "Actions", "Reset", and "View Plug-in". The main area is a table with the following data:

Name	Type	Updated	Version	About URL
AG-GRID (Offline)	Region	4 days ago	0.3	<a href="https://github.com/phartenfeller/apex-ag-grid-plugin">https://github.com/phartenfeller/apex-ag-grid-plugin</a>
Offline Data List	Region	5 days ago	0.1	<a href="https://github.com/phartenfeller/apex-off-grid">https://github.com/phartenfeller/apex-off-grid</a>
Offline Form Utils	Dynamic Action	17 hours ago	0.1	<a href="https://github.com/phartenfeller/apex-off-grid">https://github.com/phartenfeller/apex-off-grid</a>
Sync Offline Data	Dynamic Action	15 hours ago	0.1	<a href="https://github.com/phartenfeller/apex-off-grid">https://github.com/phartenfeller/apex-off-grid</a>

# Setup Sync-Plug-In

The screenshot shows a configuration interface with a sidebar and a main panel. The sidebar includes icons for file, lightning bolt, circular arrow, and triangle. The main panel has a header with 'Actions Fired on Page Load' and a settings button. Below this, under 'Events' > 'Page Load', there is a section for 'Setup offline storages' which is set to 'True'. Within 'True', a 'Sync Offline Data [Plug-In]' action is selected and highlighted with a green dotted border. Other options like 'False' and 'Change' are also shown.

The screenshot shows the configuration details for the 'Sync Offline Data [Plug-In]' action. At the top, it says 'Action' and 'Sync Offline Data [Plug-In]'. Under 'Identification', the 'Source Query' is defined as:

```
SELECT id,  
       email,  
       salary,  
       company,  
       skillset,  
       last_name,  
       first_name,  
       shirt_size,  
       birthday,  
       round(( last_changed - DATE '1970-01-01' ) * 60 * 60 * 24) * 1000 AS LAST_CHANGED  
FROM offl_people
```

Below this, several settings are listed:

- Unique Storage ID: people
- Storage Version: 1
- PK Column: ID
- Last Changed Column: LAST\_CHANGED
- Rows to fetch per request: 500
- Sync timeout minutes: 60

# Add consumer Plug-In

Region Attributes

Filter ↻

**Identification**

Title AG-Grid Offline

Type AG-GRID [Plug-In]

**Source**

Location Local Database

Type SQL Query

SQL Query

```
select ID ,  
EMAIL ,  
SALARY ,  
COMPANY ,  
SKILLSET ,  
LAST_NAME ,  
FIRST_NAME ,  
SHIRT_SIZE ,  
BIRTHDAY ,  
round(( LAST_CHANGED - date '1970-01-01' ) * 60 * 60 * 24) * 1000 as  
LAST_CHANGED  
from OFFL_PEOPLE
```

Region Attributes

Filter ↻

**Settings**

Primary Key Column ID

Focus first cell

Display Rownumber

Page Size 30

Additional Settings

```
{  
  offline: {  
    storageId: "people",  
    storageVersion: 1  
  }  
}
```

# Add save Action

The screenshot shows a configuration interface with a sidebar on the left and a main panel on the right. The sidebar has a tree structure:

- AG-Grid Offline
  - Columns
  - Copy
    - SAVE
      - Dynamic Actions
        - Save Data
          - True
            - SAVE-AG-GRID-DA (Offline) [Plug-In]
          - False

The screenshot shows the 'Action' configuration panel with the following settings:

- Action: SAVE-AG-GRID-DA (Offline) [Plug-In]
- Affected Elements
  - Selection Type: Region
  - Region: AG-Grid Offline

**Additionally**

You need to install packages and a table

You need to write Code to merge your data back to the source (more on that later)

# Demo - Basic offline data editing

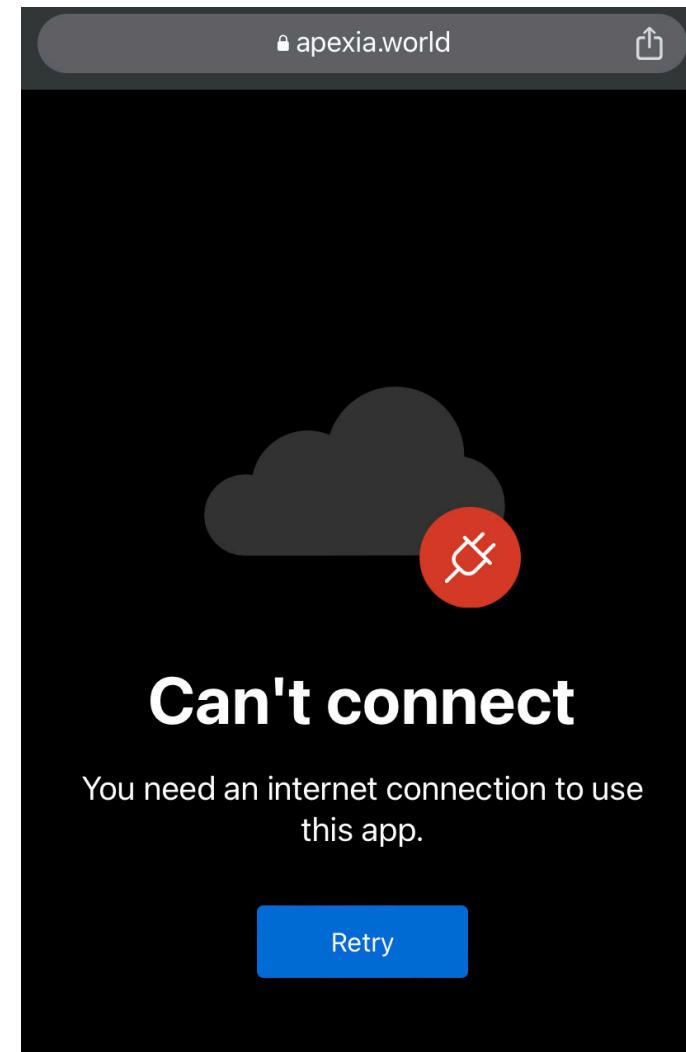


# Design Decisions



# How far can offline go

- Level 1 (APEX currently)
  - Custom Offline Page instead of Browser Error
- Level 2
  - Cache Static Content (HTML, Assets, ...)
  - In-app Navigation
  - No Interactivity (paginations, filters, save etc.)
- Level 3
  - Everything works as if you were online
  - **You need to bring the data to the client**



## Don't work against APEX

- APEX Components are designed to work with a live-connection to the Database
  - It is not trivial to hook into this and **APEX is not designed to do this**
  - I don't want to mess with that
- Own Plug-Ins where I have full control

# SQL + Relational

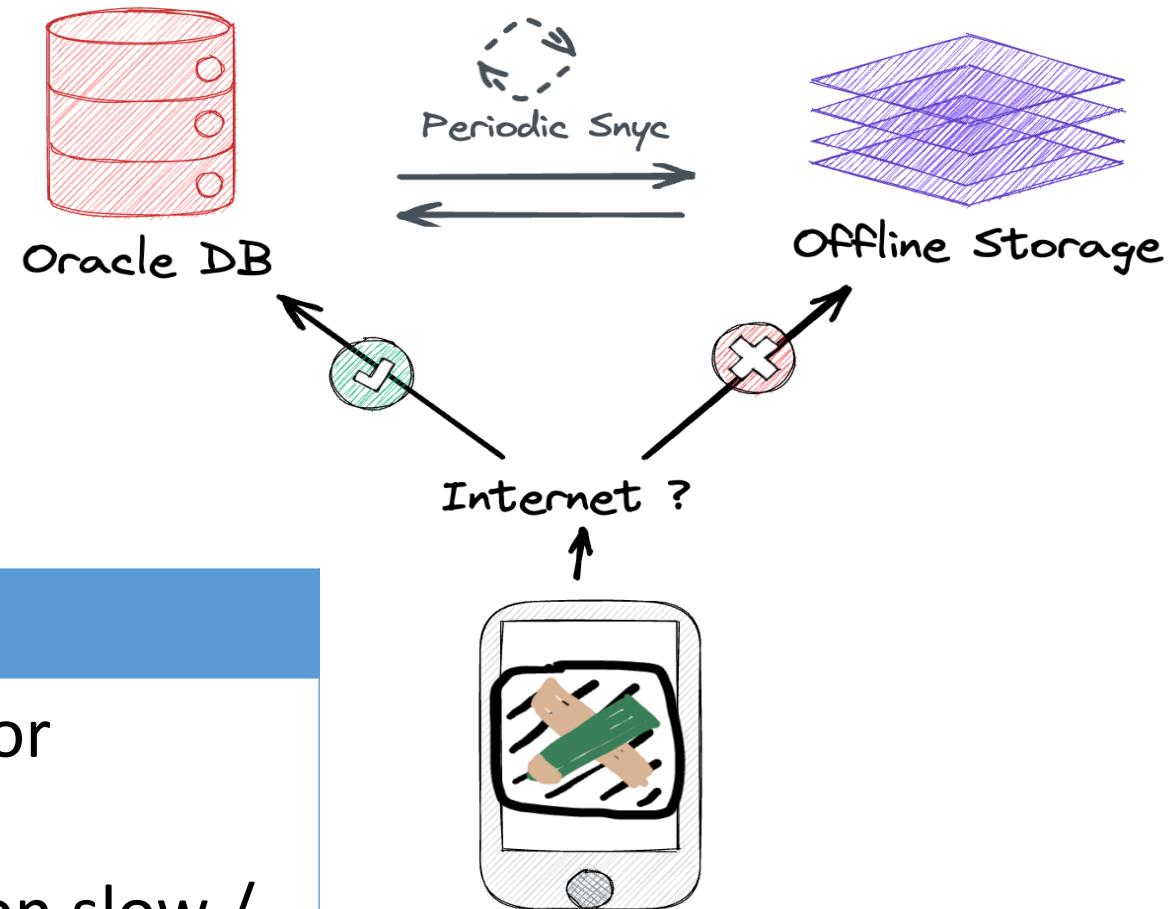
- Same data storage as the central database
- Reusability
- Performance
- Analytical functions
- Data Integrity



Original Photo by [Eye for Ebony](#) from [Unsplash](#)

# Data Access Strategies - Online First

- First try to get data from the DB
- Only when no connection available use offline storage



## Pros

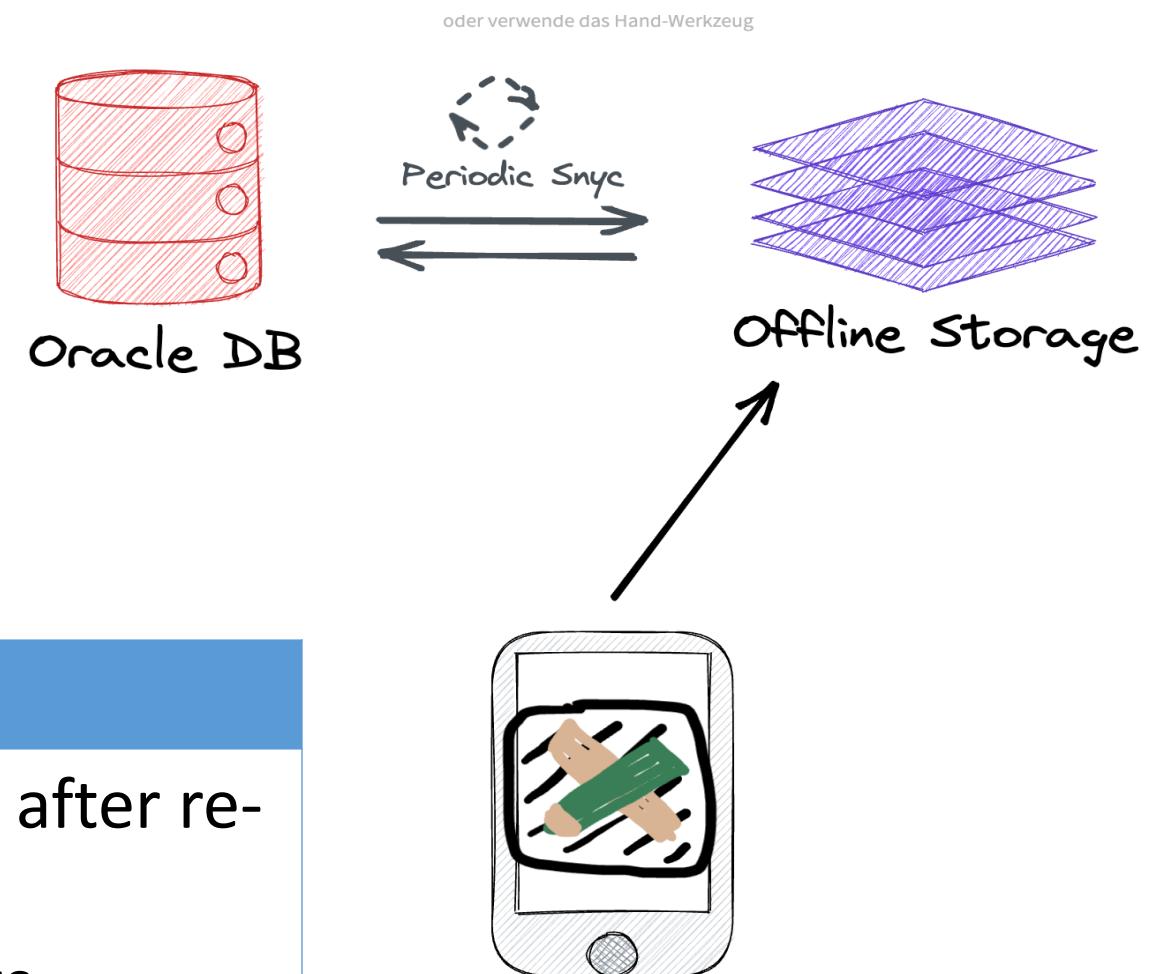
- Online = “Normal APEX Mode”
- Less conflicts
- Live data

## Cons

- Much logic for consumers
- What to do on slow / unreliable connections

# Data Access Strategies - Offline First

- Always use Offline Storage for your components



Pros	Cons
<ul style="list-style-type: none"><li>• Reliable Experience</li><li>• Fast</li><li>• Less logic for consumers</li></ul> <p><i>I chose this approach because of this</i></p>	<ul style="list-style-type: none"><li>• Only updates after re-sync</li><li>• More conflicts</li></ul>



## Build different experiences for different use cases

- Make sure only people who need offline access use it
- Field technician gets offline App / Page
- Office worker gets online App / Page



Original Photo by [MIOPS Trigger](#) from [Unsplash](#)

# Technical Implementation



# Where to store data

Browser Storage Options:

- LocalStorage
  - Key-Value object-store, slow, designed for small amounts
- WebSQL
  - Deprecated
- IndexedDB
  - For large data, transactions, Key-Value object-store

## The thing with IndexedDB:

- Complex API
- No support for complex queries
- History of nasty bugs (mostly Safari)
- Not ACID compliant

[The pain and anguish of using IndexedDB: problems, bugs and oddities](#)

# Not the DB we deserve, but the hero we need

The image shows a screenshot of a Twitter post. At the top left is the Chromecast icon. Next is the profile picture of the account, which is the Google Chrome logo. The account name is "Chrome Developers" with a verified badge. Below the name is the handle "@ChromiumDev". To the right is a three-dot ellipsis menu. The tweet text reads: "Yes, the plan is to eventually remove Web SQL, but 🥁 our intention is to empower developers to create their own solutions for structured storage, and we're therefore working with the #SQLite team to create a SQLite implementation over Wasm. This solution will replace Web SQL 💪!". Below the text is the timestamp "12:33 AM · Sep 1, 2022".

Chrome Developers Verified  
@ChromiumDev

Yes, the plan is to eventually remove Web SQL, but 🥁 our intention is to empower developers to create their own solutions for structured storage, and we're therefore working with the #SQLite team to create a SQLite implementation over Wasm. This solution will replace Web SQL 💪!

12:33 AM · Sep 1, 2022

[Source](#)

## Wait a minute... What is SQLite?

- Most used DB, top 5 most deployed software ever (estimated one trillion active DBs)
- Full-featured SQL
- Serverless (not the cloud thing)
- Transactional
- ACID
- Public Domain
- Really fast
- Stores data in a single file
- Recommended Storage Format by the US Library of Congress

[More on SQLite](#)

# How does it run in the browser



## WEBASSEMBLY

Compile existing software to hardware-near browser understood layer

### Origin Private File System (OPFS)

- Sites get a private file system
- SQLite directly writes to here

Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
86	86	111	72	15.2	109	111	74	15.2	No	109

[Source](#)

## Safari Again...

- Currently a bug in iOS 16
- Won't work at all
- Even though it is fixed in Safari Tech Preview since **months**, it seems like they only ship this fix with Safari 17



Original Photo by [Nathan Dumlao](#) from [Unsplash](#)

# How fast is it ? - Demo

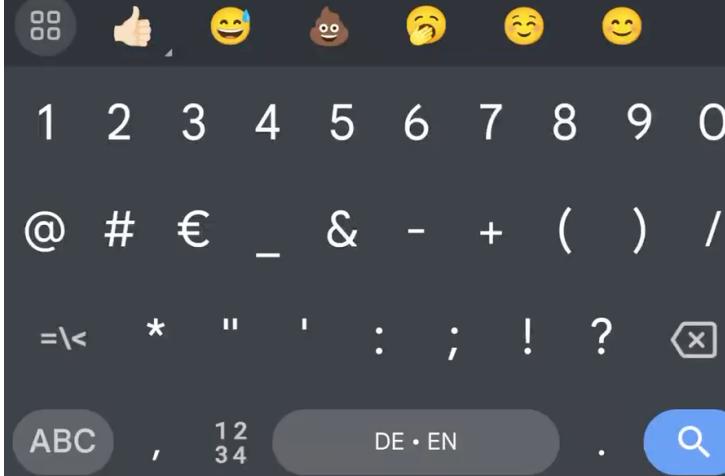


On a Phone?  
(Pixel 6)

## Movies

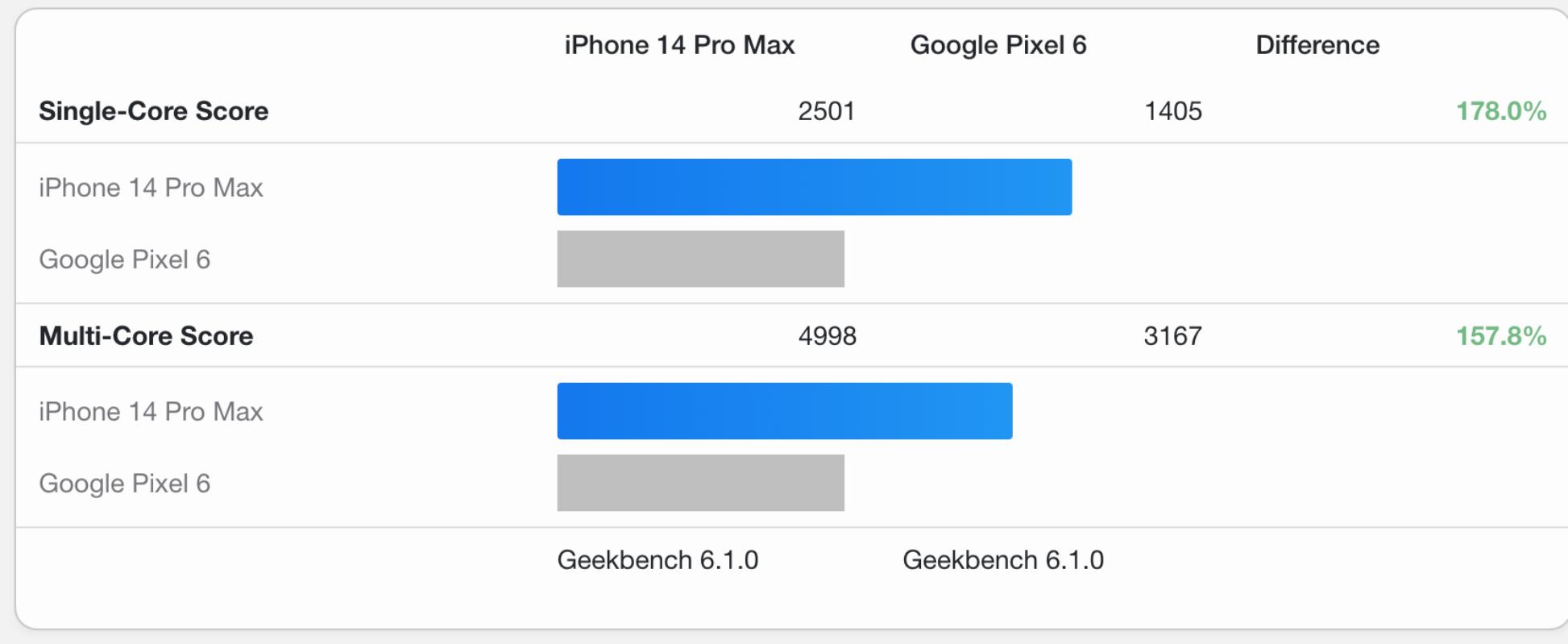
## New

	Orig Title	Year
Movie Id	Filter by Orig Tit	Filter by Year...



# Context

## iPhone 14 Pro Max vs Google Pixel 6



<https://browser.geekbench.com/v6/cpu/compare/1716873?baseline=1715979>

# How develop Plug-Ins

```
> window.hartenfeller_dev.plugins.sync_offline_data.storages.movies_v1
< ▶{isReady: true, config: {...}, getColInfo: f, getRowByPk: f, getRows: f, ...} ⓘ
  ►config: {pageSize: 500}
  ►getColInfo: () => _getColInfo(storageId, storageVersion, apex)
  ►getRowByPk: (pk) => _getRowByPk(storageId, storageVersion, pk, apex)
  ►getRowCount: ({ searchTerm, colFilters }) => {...}
  ►getRows: ({ offset = 0, maxRows = 100, orderByCol, orderByDir, searchTerm, colFilters, getR
    isReady: true
    ready: true
  ►sync: () => {...}
  ►writeChanges: (rows) => {...}
  ►[[Prototype]]: Object
```

# Data Synchronization



## Why this topic is important

- Allowing users to edit data without a connection to the DB has effects of **decentralization**
- You still want a single source of truth  
-> This results in new challenges for data consistency

## Sync Timeline - 1

1. Cache current HTML page
2. Synchronize Client Changes with DB
  - Send client updates to central DB
  - Delete corresponding rows on the client
  - It is now in the control of the DB what will happen with these rows

## Sync Timeline - 2

### 3. Synchronize DB Changes with Client

- DB sends PK + Timestamp for each row:
  - PK known and Timestamp matches: 
  - PK unknown or Timestamp mismatch:  request full row -> insert / update
- Delete rows not present in central DB anymore:
  - Store every PK received from server in a temp table
  - After receiving all rows, delete local rows where PK not in temp table

# Change Synchronization - Potential outcomes

## a: Single Client Row Update (✓)

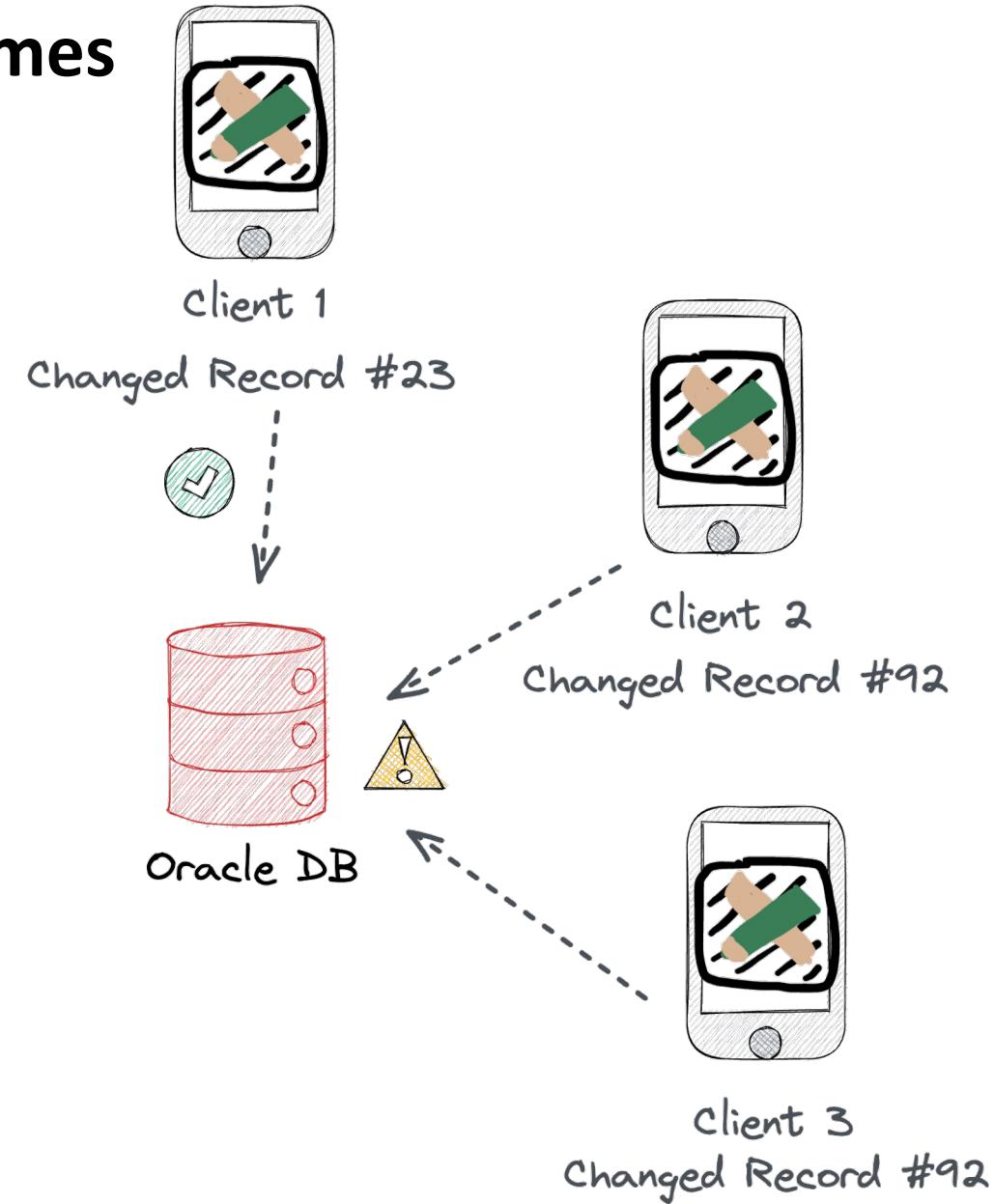
- Only one client modifies the row, straightforward synchronization

## b: Sequential Client Row Updates (✓)

- Client 2 updates row after receiving Client 1's changes, preserving data integrity

## c: Concurrent Client Row Updates (!)

- Two clients modify the same row version simultaneously, potential data conflict



# Conflict Resolution #1 – Last Write Wins

Design Decision



Nothing to consider:  
Just write changes

Photo by [Matthias Speicher](#) from [Unsplash](#)



Photo by [Ryan Moreno](#) from [Unsplash](#)

## Conflict Resolution #2 – Merge Conflict Detection

### Changes Table

ChngType	Data	RowSrcTimeStamp	RowChngTimeStamp	...
U	{...}	192....0003	192....5843	



= src table row timestamp?  → update src table row timestamp now 192....5843

# Conflict Resolution #2 – Merge Conflict Detection

## Changes Table

ChngType	Data	RowSrcTimeStamp	RowChngTimeStamp	...
U	{...}	192....0003	192....5843	



= src table row timestamp?  → update src table row timestamp now 192....5843

Later...

U	{...}	192....0003	192....9932
		↓	

src table row ts 192....5843 != 192....0003 

→ changes need to be manually merged

## How changes are stored

- Don't directly insert into source table to handle conflicts
- Generic changes table

Storage ID Storage Version	Data JSON Change Type	Timestamp Src Timestamp Chng	Audit Cols	Import Failed
people 2	{ ... } I, U or D	192....0003 192....9843	JAMESBOND 00716(Session)	0 Someday False :D (23c)
...				

# Demo – Merge Code



# How to do a manual merge

- On conflict: failed col set to 1
- User must do merge manually
  - Show current row next to changed row (like Git 3-way-merge)

192...0003

Firstname: Peter

Lastname: Parka

Age: 23

192...5843

Firstname: Peter

Lastname: Parker

Age: NULL

After Merge Row

Firstname: Peter

Lastname: ???

Age: ???

....

....

....

Let us create chaos

<https://tinyurl.com/kspc23-off>



Does not  
work on iOS!  
(bug)

# Wrapping Up



# What can be improved

- Data type handling
  - Dates, Timestamps ...
  - BLOBs
- LOVs (Select Lists, Popup LOV, Shuttle, Checkboxes...)
- Apply more table rules to client (FKs, check constraints, indexes, defaults...)
- Error message improvements and error handling
- Expand Plug-In Ecosystem
- Use JSON-Duality features (only 23c)

# Why you may not want to use this

- You lose authority over your data
  - Data on the client can easily be duplicated / shared
- Merge conflicts / Inconsistent data state
- Divergent development approach for offline apps
- Increased complexity
- Scalability: Synchronization processes can demand significant data transfers
- SQLite as fast as the device using it
- Proof of Concept + No Support

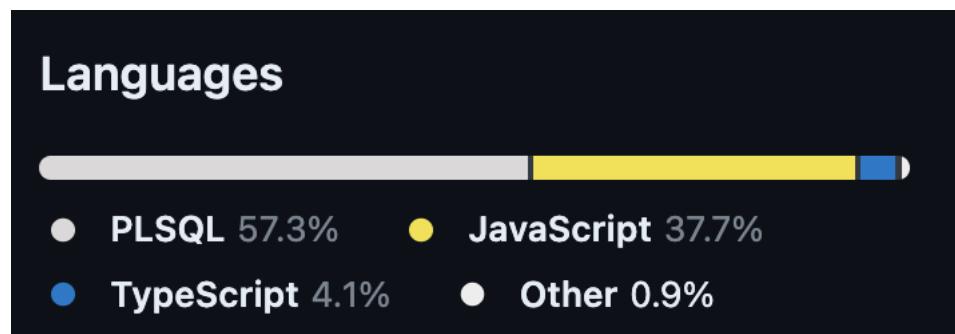


Photo by [JESHOOTS.COM](#) from [Unsplash](#)

# Where can I get this ?

<https://github.com/phartenfeller/apex-off-grid>

- Needs Documentation
- Needs Refactoring
- Contributions welcome 😊



# Conclusion

- I am quite happy with the result
- Plug-Ins easy to use, hide complexity well
- Confident that SQLite is a great choice for small to large data sizes
- Great browser support (**without Safari bug**)



Photo by [Sebastian Herrmann](#) from [Unsplash](#)

# Testing APEX Apps is now as easy as creating them.

- Tailored to APEX
- Save a lot of time on regression tests
- Use our intuitive LCT-App and don't write any test code
- Testing on multiple platforms simultaneously



LCT



You heard  
it here first!  
#kscope23

# TEMPLATE COMPONENT **PLUGIN** **CHALLENGE**

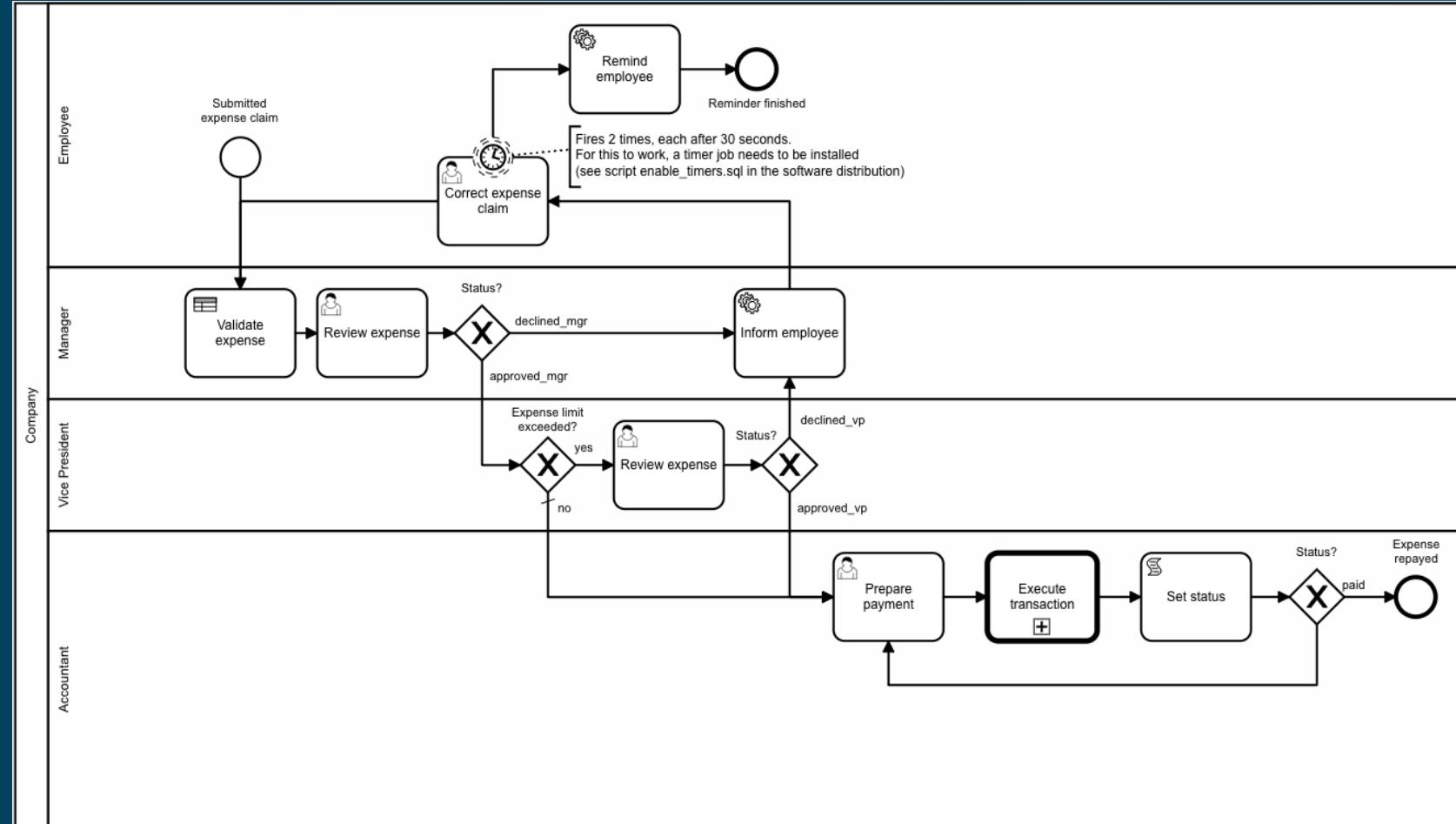


**August 2023**  
Subscribe to get notified!



# Flows for APEX

## BPMN 2.0 Workflows for APEX



- Open Source
- Community Driven
- Support available



The Educational Conference For Oracle Technology Users

ODTUG  
**Kscope23**  
aurora, co      june 25-29



Interested in volunteering?  
Scan the code to sign up



Don't Forget To  
Fill Out Your Evals