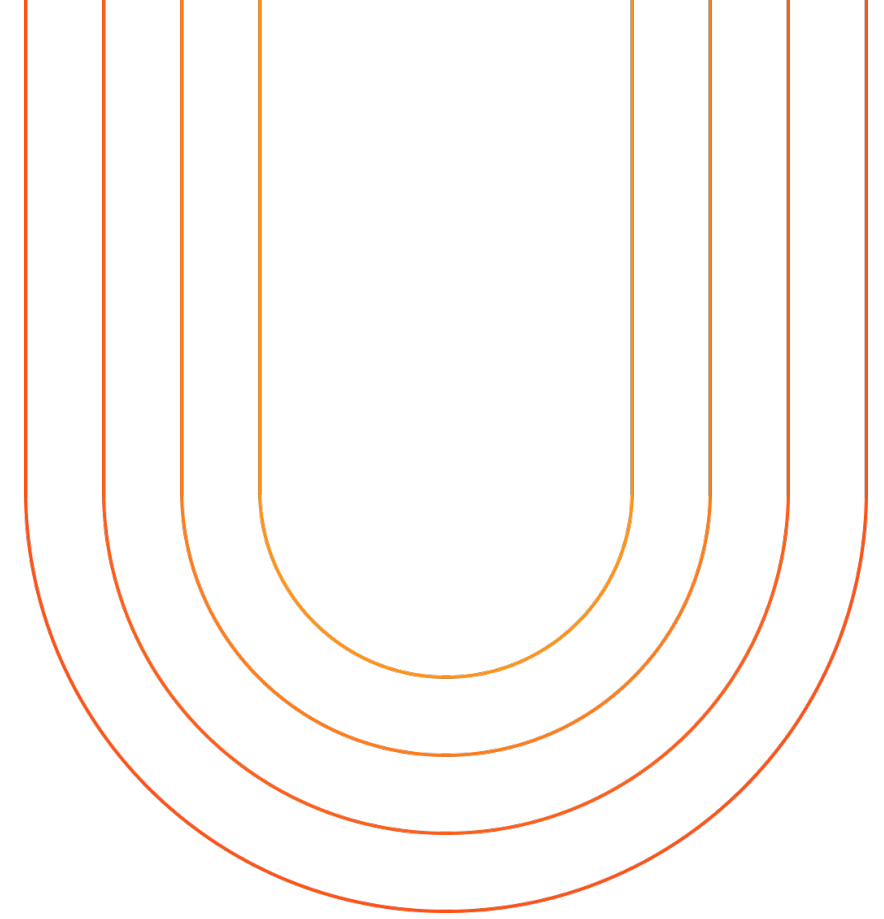


# UI5 Web Components

## using React and Serverless first

Thiago Sasai, Birlasoft  
July 7<sup>th</sup>, 2022



# Agenda

What we'll build

Some words about Serverless

Technical Architecture

Frontend

Backend

Let's explore the code

# What we'll build

UI5con HYBRID<sup>®</sup>

CRUD with UI5 Web Components for React - EN

+

2,000.00

Total Credit

-

661.00

Total Debit

\$

1,339.00

Balance

Transaction List

New Transaction

Date	Payee	Category	Payment Method	Cash Flow	Amount	Actions
Jun 20, 2022	Initial balance	Wage	Bank	CREDIT	2,000.00	<div><div></div><div></div></div>
Jun 20, 2022	Gas Station	Car	Mastercard	DEBIT	235.00	<div><div></div><div></div></div>
Jun 22, 2022	Restaurant	Meal	Cash	DEBIT	65.00	<div><div></div><div></div></div>
Jun 22, 2022	Breakfast Shop	Meal	Cash	DEBIT	15.00	<div><div></div><div></div></div>
Jun 24, 2022	Supermarket	Market	Credit	DEBIT	346.00	<div><div></div><div></div></div>

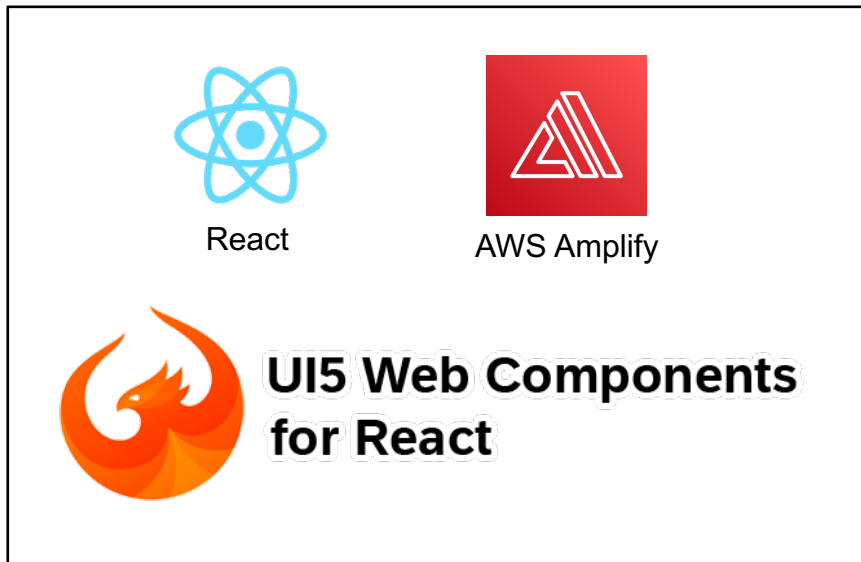
# Some words about Serverless

Serverless is a cloud-native development model that allows developers to build and run applications without having to manage servers.

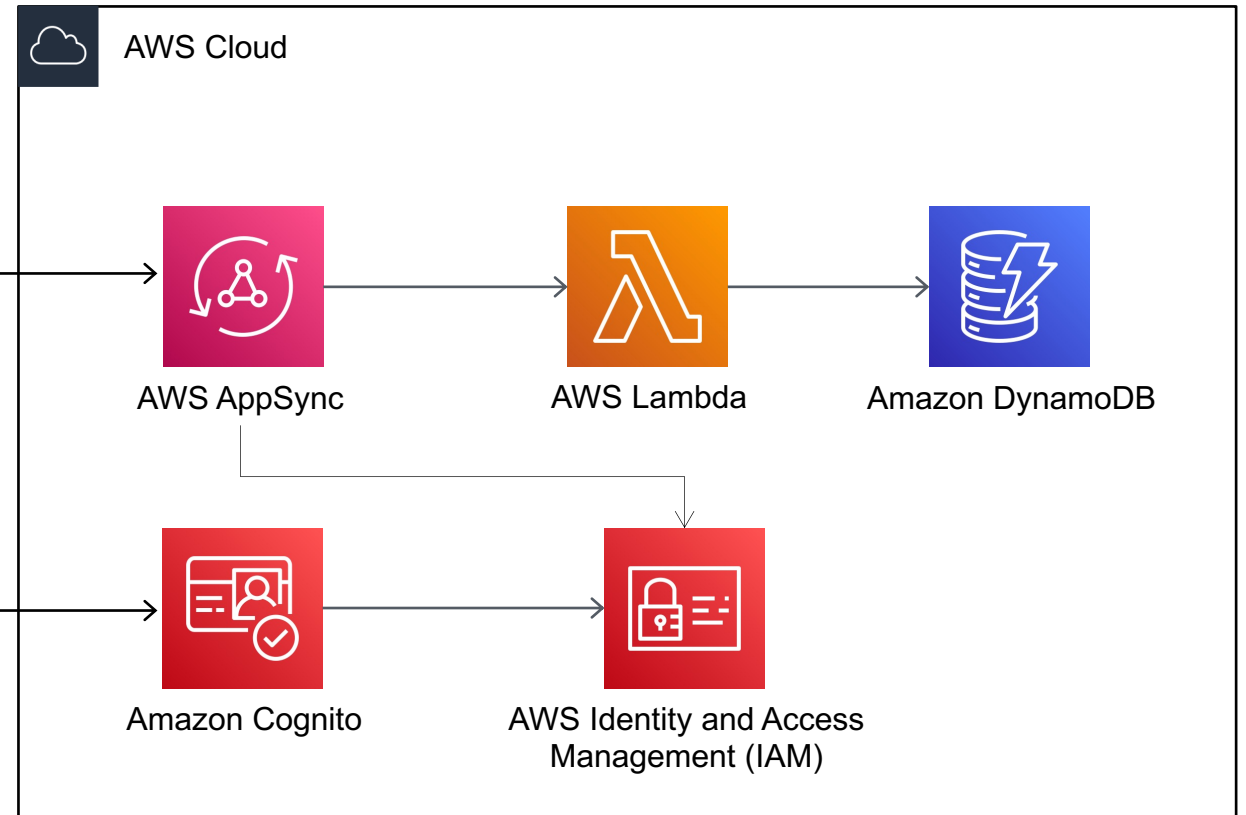
Of course, there are still servers in serverless, but they are abstracted away from application development, all routine related with provisioning, maintaining and scaling is handled by a cloud provider.

# Technical Architecture

## Frontend



## Backend



# Frontend

# Create a new React app

## Create a React app using UI5 Web Components for React template

```
> npx create-react-app ui5con-crud --template @ui5/webcomponents-react  
> cd ui5con-crud  
> npm start
```

## Install UI5 Web Components React Charts (it is not installed by template)

```
> npm install @ui5/webcomponents-react-charts
```

## Install Amplify libraries into React app

```
> npm install aws-amplify @aws-amplify/ui-react
```

## Install React Router libraries

```
> npm install react-router-dom@6.3.0
```

**Backend**





# Initialize a new backend

Install and configure the Amplify CLI (<https://www.youtube.com/watch?v=fWbM5DLh25U>)

```
> amplify init
```

```
? Enter a name for the project ui5concrud
The following configuration will be applied:

Project information
| Name: ui5concrud
| Environment: dev
| Default editor: Visual Studio Code
| App type: javascript
| Javascript framework: react
| Source Directory Path: src
| Distribution Directory Path: build
| Build Command: npm.cmd run-script build
| Start Command: npm.cmd run-script start

? Initialize the project with the above configuration? (Y/n) Y_
Using default provider awscloudformation
? Select the authentication method you want to use: (Use arrow keys)
> AWS profile
  AWS access keys
```

# Create GraphQL API and DynamoDB table

## Add API to create, read, update and delete Transactions items:

```
> amplify add api
```

```
? Select from one of the below mentioned services: GraphQL
? Here is the GraphQL API that we will create. Select a setting to edit or continue Continue
? Choose a schema template: Blank Schema

⚠ WARNING: your GraphQL API currently allows public create, read, update, and delete access to all models via an API Key. To configure PRODUCTION-READY authorization rules, review: https://docs.amplify.aws/cli/graphql/authorization-rules

✔ GraphQL schema compiled successfully.

Edit your schema at [REDACTED]\ui5con-crud\amplify\backend\api\ui5concrud\schema.graphql or place .graphql files in a directory at [REDACTED]\ui5con-crud\amplify\backend\api\ui5concrud\schema
✔ Do you want to edit the schema now? (Y/n) · yes
Edit the file in your editor: [REDACTED]\ui5con-crud\amplify\backend\api\ui5concrud\schema.graphql
✔ Successfully added resource ui5concrud locally

✔ Some next steps:
"amplify push" will build all your local backend resources and provision it in the cloud
"amplify publish" will build all your local backend and frontend resources (if you have hosting category added) and provision it in the cloud
```

# Define schema for GraphQL API

Edit file `ui5con-crud/backend/api/ui5concrud/schema.graphql`

```
3  input AMPLIFY {  
4    globalAuthRule: AuthRule = { allow: public }  
5  } # FOR TESTING ONLY!  
6  type Transaction @model {  
7    id: ID! @primaryKey(sortKeyFields: ["date"])  
8    date: AWSDate!  
9    payee: String!  
10   category: String!  
11   paymentMethod: String!  
12   cashFlow: CashFlow!  
13   amount: Float!  
14   comment: String  
15  }  
16  
17  enum CashFlow {  
18    debit  
19    credit  
20  }
```

# Push to the Cloud

Push to the cloud, it will create the API into AppSync, the DynamoDB table and all Lambda Functions and resolvers to link API with Database

> amplify push

Category	Resource name	Operation	Provider plugin
Api	ui5concrud	Create	awscloudformation

```
? Are you sure you want to continue? Yes
? Do you want to generate code for your newly created GraphQL API Yes
? Choose the code generation language target javascript
? Enter the file name pattern of graphql queries, mutations and subscriptions src\graphql\**\*.js
? Do you want to generate/update all possible GraphQL operations - queries, mutations and subscriptions Yes
? Enter maximum statement depth [increase from default if your schema is deeply nested] 2
- Updating resources in the cloud. This may take a few minutes...
```

# Add Authentication

## Create Authentication Service

```
> amplify add auth
```

```
The current configured provider is Amazon Cognito.  
  
Do you want to use the default authentication and security configuration? Default configuration  
Warning: you will not be able to edit these selections.  
How do you want users to be able to sign in? Username  
Do you want to configure advanced settings? No, I am done.
```

## Push to the Cloud? It will provision Cognito service

```
> amplify push
```

Category	Resource name	Operation	Provider plugin
Auth	ui5concrudcc51db7e	Create	awscloudformation
Api	ui5concrud	No Change	awscloudformation

```
? Are you sure you want to continue? Yes
```

# Add UI5 Web Components for React

# Add frontend code

**Copy the content of files below from git repository to your project.**

- Git repository: <https://github.com/tsasai7/ui5con-crud.git>
- Files:
  - **public/UI5con\_logo.png**: logo to display into ShellBar.
  - **public/index.html**: add line 32 ("fetchDefaultLanguage": true) it is required to language identification (i18n).
  - **public/i18n/i18n\_en.properties**: English translation.
  - **public/i18n/i18n\_pt.properties**: Portuguese translation.
  - **src/index.js**: register i18n (internationalization) properties files and render the React app.
  - **src/App.js**: define UI5 theme and route paths using React Router.

# Add frontend code – React Components

Copy the content of files below from git repository to your project.

## ■ Files:

- **src/components/Main.jsx**: define main page, with ShellBar and FlexBox to display page contents.
- **src/components/ProductPopover.jsx**: define Product popover of ShellBar (🏠) with navigation Dashboard and Transactions pages.
- **src/components/ProfilePopover.jsx**: define Profile popover of ShellBar (👤) with Logout action.
- **src/components/Dashboard.jsx**: define Dashboard page, example of PieChart component usage.
- **src/components/Transactions.jsx**: define Transaction page with BalanceBox, Panel and TransactionTable components.
- **src/components/TransactionTable.jsx**: usage of AnalyticalTable.
- **src/components/TransactionDialog.jsx**: usage of Dialog with Form to create and edit entries.
- **src/components/BalanceBox.jsx**: usage of Card to display balance boxes (total credit, total debit and result).



**Let's Explore the Code**

# Thank you!

**Thiago Sasai**  
SAP Practice Manager  
twitter: @tsasai7