AvailsBridge Road Map | Leandro Silva, Tremor Video, 2014
All the below logic happens daily or whenever the app is opened.

Legend:
- Server data / CRON

AvailsBridge

Start Here

Config File

- Rate Type Calculation
  - I.e. CPE = Quantity/[Performance]
  * A word in brackets tells the app to look in the config file for data, otherwise it looks in the RFR Report.
- Performance
  - I.e. type = price
- Product Group
  - I.e. sub group = group
- Projected delivery in upcoming months
  - 12 months from current month with values attached to each month.
- Cap ratio
- Hair Cut ratio

Store config data to a master file online:

- When admins edit the config file, the data gets pushed to a server
- When the app is opened, the local config file gets replaced with the master config file

RFR Report
(SFDC data)

Column data needed:

- GEO A + GEO Z
- Line Item ID
- DaysLive
- Start Date
- End Date
- Rate Type
- Product Name
- Quantity

DMA Mapping Table

Map city to DMA code and map related state-DMAs to states and DMAs

Output JSON:

- cityToDMA
- relatedStates
- relatedDMAs

OLAP

Column data needed:

- State Code
- DMA
- Total

*Run a query to get server data from the last 30 days for State, DMA, and Impression on US PreRolls where the campaign type is not 'self serve' nor 'integration.'

Database

Projected state-DMAs inventory:

(([OLAP total column for each row] / [grand total of all OLAP total column]) * [projected delivery data from Config]) / [total days in month]

Store value in DB: ([value from above] * [cap ratio from Config]) * [hair cut ratio from Config]

*For each month of the projected year

Inventory pool Table:
(Inventory of total network based on projections and historic data)

- State-DMAs: String
- Date: String
- Projected state-DMAs inventory: Number

Updates daily (CRON) by recalculating with OLAP data.

*A row for every state-DMAs
*A column for every month of the projected year

True impression calculation:
From Config + RFR data

Use RFR and Config data to convert unit goals to impressions (if not already impressions). This is done for every row in the RFR Report:

- Grab data from the RFR's 'Rate Type' column
- Grab data from the RFR's 'Product Name' column'
- Determine the calculation based on the Config File's data and the specified RFR 'Rate Type'
- Determine the product group based on the Config File's data and the specified RFR 'Product Name'
- Search the Config File for a matching calculation based on the values determined by RFR and Config data
- Parse the above values as the specified 'Rate Type Calculation' string as math

Output payload:
First user of the days kicks off this process. Every user after grabs the daily file.

- Line Item Number
- GEOs
- True impression
- Days Live
- Start Date
- End Date
- Push mapped DMA code to city values
- State-DMAs

State-DMAs:

- Get every state-DMAs related the GEO(s)
- E.g. If a row has the state AZ, pull the state-DMA mapping table's related states/DMAs

Store payload in intranet file

Initial UX:

- User enters start and end date
- User chooses to search via state/DMA
- User chooses from state/DMA drop down (depending on last step)
  *UI allows the user to add multipe states/DMAs
- User clicks 'calculate' button

Get list of related state-DMAs
List based on user inputted state/DMA

Check 'Booked amount for state-DMAs' table for a non-zero cell with the specified state-DMAs item on each of the specified date

If data exists in requested cell

If data does not exists in requested cell

Booked amount for state-DMAs Table:
(Amount booked for each inventory pool for every day of the year)

This table to expedite searchs: if day value exists, use it

- State-DMAs: String
- Date: String
- Avails: Number

*A row for every state-DMAs
*A column for every day of the year

*Cleared on new RFR data

Booked amount for state-DMAs at specific date:

If conditions are met:
- User selected start date is less than payload end date
- User selected end date is greater than payload start end
- User selected state/DMA must have at least one match from payload state-DMAs
- E.g. [User selected state array] must having matching index to [payload state-DMAs array]

Returns a list of L# from the payload that meet the above

For each L# + each state-DMA in that L# + every date from the user selected date range, do:

L# 1 Inventory Pool Table value / Inventory Pool Table sum of values
((([matching state-DMAs item] / [total of all state-DMAs in array]) * [true impression per row]) / [Days Live]) - [matching state-DMAs item]
((([next matching state-DMAs item] / [total of all state-DMAs in array]) * [true impression per row]) / [Days Live]) - [next matching state-DMAs item]
...and so on

L# 2
...same as above, but sum previous overlapping state-DMAs on matching dates (updates relevant daily amount)

Saves above 'daily amount' to Booked amount for state-DMAs table

Avails:

Sum all returned cell data and output final result