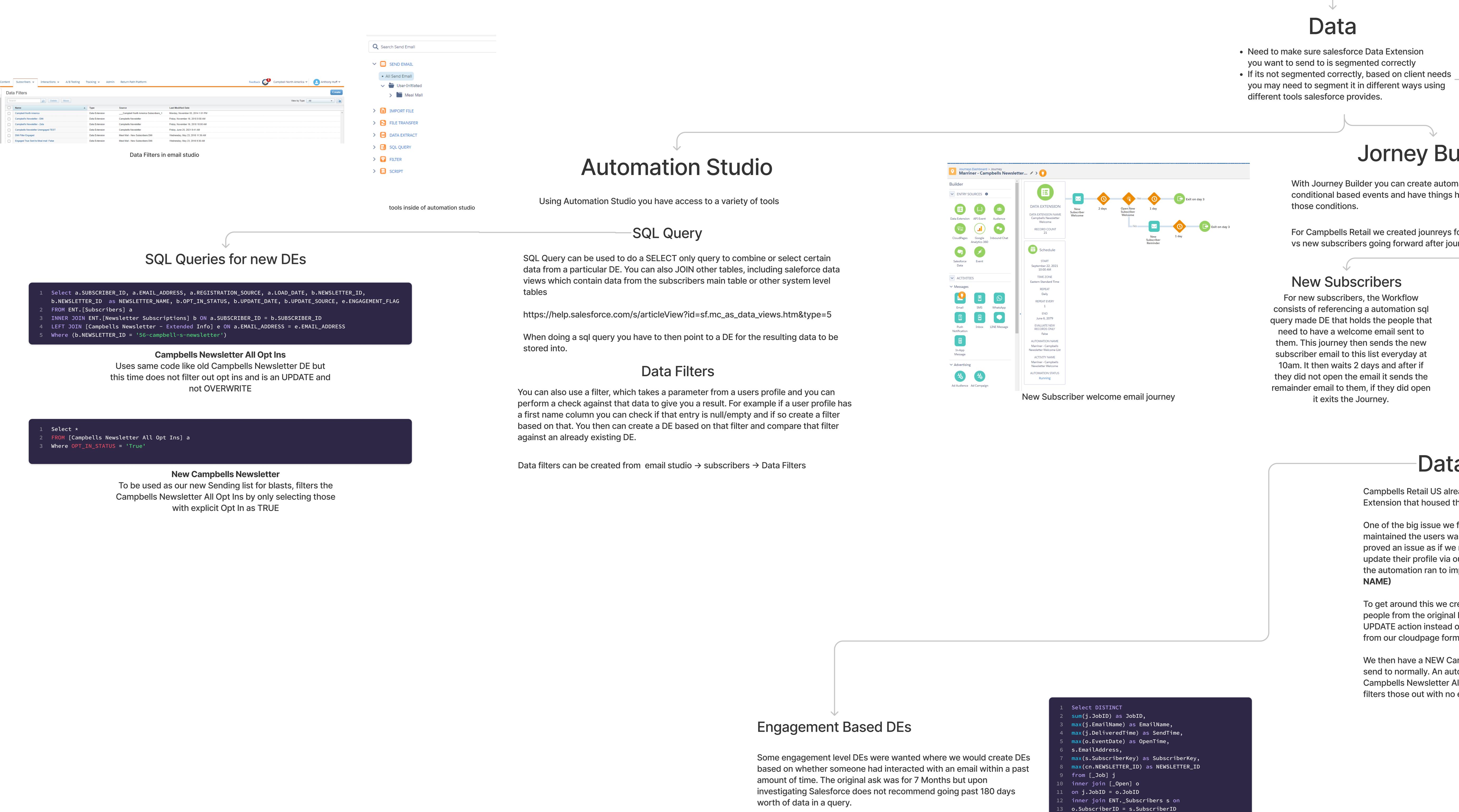
Salesforce Process Outline



Audience 0 100κ 200κ 300κ 400κ 500κ 600κ 700κ

Selective Subscriber

Window Shoppe

Dormant/Winback

Jorney Builder

With Journey Builder you can create automatic situations that take conditional based events and have things happen based on outcomes of those conditions.

For Campbells Retail we created jounreys for users that would be based on current subscribers vs new subscribers going forward after journeys have started

New Subscribers

For new subscribers, the Workflow consists of referencing a automation sql query made DE that holds the people that need to have a welcome email sent to them. This journey then sends the new subscriber email to this list everyday at 10am. It then waits 2 days and after if they did not open the email it sends the remainder email to them, if they did open

it exits the Journey.

inner join [NEW Campbells Newsletter] cn on

cn.NEWSLETTER_ID = '56-campbell-s-newsletter' and

Opens in the last 180

s.SubscriberKey = CONVERT(varchar(100),

j.EmailName NOT LIKE '%CCNW%' and

cn.EMAIL_ADDRESS as EmailAddress,

FROM [NEW Campbells Newsletter] cn

ON op.SubscriberKey = cn.SUBSCRIBER_ID

No opens in last 180 days

EFT JOIN [OpensIn180Days] op

WHERE op.SubscriberKey IS NULL

CONVERT(varchar(100), cn.SUBSCRIBER_ID) as

1 GROUP BY s.EmailAddress

SubscriberKey

others. They break them into 4 catergories. You can reference this via

0 o.EventDate > dateadd(d,-180,getdate())

-Data Extensions-

Existing Subscribers

Campbells Retail US already had a large over a million subscriber Data Extension that housed the users they would blast to for emails.

One of the big issue we found with this DE was that Automation that maintained the users was set to OVERWRITE and not UPDATE which proved an issue as if we recreated this DE to use if a user would update their profile via our forms the new data would be wiped once the automation ran to import users via the third party service (NEED

To get around this we created a new starting DE to house the same people from the original DE script import but this time its using a UPDATE action instead of OVERWRITE, we use this new DE to write to from our cloudpage forms. (Campbells Newsletter All Opt Ins)

We then have a NEW Campbells Newsletter DE that we will use to send to normally. An automation runs data into this DE from the Campbells Newsletter All Opt Ins where opt ins are equal to True so it filters those out with no explicit opt in.

Import from 3rd party (GOES into all subscribers)

new DE Data Flow for CFR Newsletter

Campbells Newsletter All Opt Ins (UPDATE) WHERE NEWSLETTER_ID = '56-campbell-s-newsletter'

NEW Campbells Newsletter (OVERWRITE)

Emails

for their corresponding group via a link short cut (%%=CloudPagesURL({cloudpage ID})=%%) where you paste the ID of the cloudpage.

When users are sent the email via an actual blast (does not work with a test send) the cloudpage

Emails can also feature in email forms. The way we currently use in CFR emails is a question based selection. We can achieve this with an actual form in the email by adding a form to the created email and selecting which DE to capture the data to and:

 Adding a field for where the data should go to. a. For example which food do you like the most?

This field would be assigned to the RECIPETYPE column in the DE • You then can assign options to this field, in this case it would still be images like we usually do but the

option is given a value so that the DE knows what to fill out that column with.

 Hidden Fields

56-campbell-s-newsletter Hidden fields needed, as they are required for the DE

can be custom built via HTML, JS, CSS. For CFR we had 2 main landing pages and their respetive forms on each page and once the form was submitted user would be directed to a

Custom code had to be done on these pages called AMPScript which is a scripting language that can be used in emails. We were able to use this to make sure the correct user was being updated when the user clicked to the cloudpage from the form as we made sure to grab their ID from salesforce and made sure to pass it in the data that was submitted.

We were also able to pre populate some text fields with data already garnered in the system if it existed for that user.

Options for this input render differently in Gmail. To learn more, see Interactive Email Display FAQs

Hover State Image Selected State Image

* Data Attribute 🕕

RECIPETYPE

Required Field

Supporting text below label

Chick Images
Replace De

Description

Image Options

* Field Value 📵

%%=v(@subscriberkey)=%% EMAIL ADDDECC set @fname = Lookup("Campbells Newsletter All Opt Ins", "FIRST_NAME", "EMAIL_ADDRESS", @email)

Required

Name of the Data Extension from which to return a value

Additional value that identifies the row to retrieve (see note)

Name of the column from which to return a value

Value that identifies the row to retrieve

To display on page if data exists:

1 set @subscriberkey = AttributeValue("_subscriberkey")

Lookup(1,2,3,4,[5a,5b]...)

2 set @email = emailaddr

1 %%[if not empty(@fname) then]%% var fnameinput = document.getElementById("fname"); fnameinput.value = "%%=v(@fname)=%%" %%[endif]%%