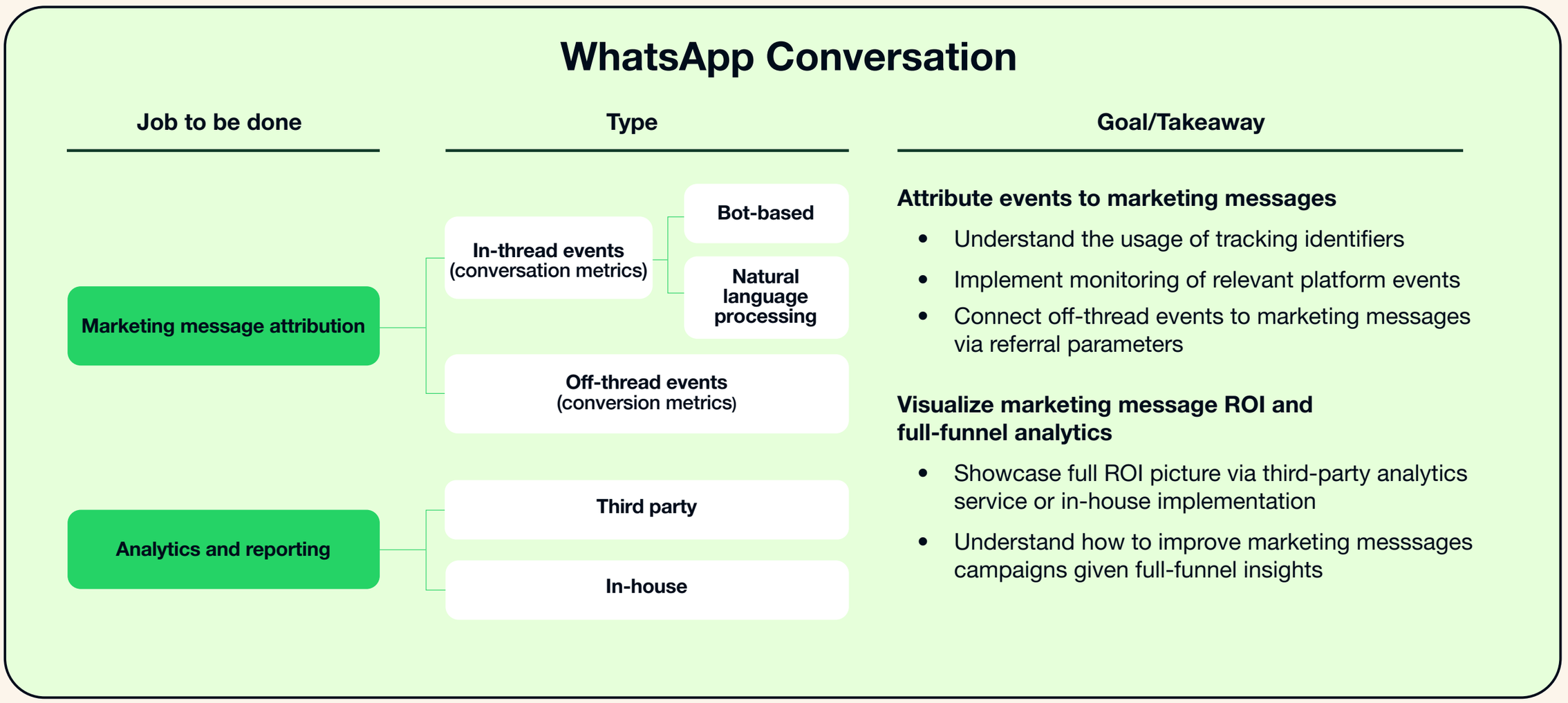
ROI measurement playbook

WhatsApp

from Meta

Chapter 1: WhatsApp marketing message attribution

To build the measurement report, we need to associate the conversation started via a WhatsApp marketing message campaign and the business outcome events it produced. Correctly logging and associating events can enable a business to populate a dashboard that clearly demonstrates the ROI of marketing messages.



Marketing message attribution

Building an attribution table

There are a few key pieces of information that need to be tracked in order to attribute conversion events to marketing messages:

Message template name	Phone number	Read timestamp	Event timestamp
Defines the campaign to attribute the events to	Identity of the user that took an action; this can be replaced with other identity information you may already have	The time at which the user read the message template	The time at which a predefined event happened for a user

The events to be tracked will differ depending on use case, but as an example, the below logging flow can be considered for commerce use cases.

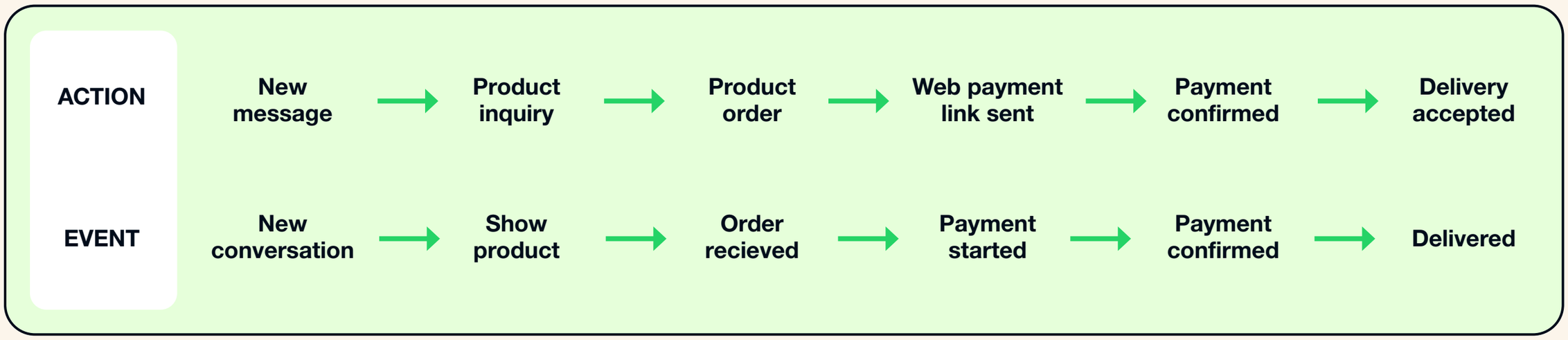
1. When a campaign is started, log the message template name and phone number of who the message was sent to. The first two columns should be filled for every phone number the marketing message has been sent to.
2. As users are reading the messages, log the timestamp of the read events for the corresponding phone number.
3. When a user makes a purchase order in-thread, log the timestamp of the purchase event.
4. When a checkout link is sent to the user, log the timestamp of the checkout_link_sent event.
5. When a website or app purchase has happened, log the timestamp of the purchase event (more on off-thread event attribution below).
6. Once all of the events are logged, you may have a table like this:

Message template name	Phone number	Read timestamp	In-thread purchase timestamp	Checkout link sent timstamp	Off-thread purchase timestamp
mega_sales_tshirts	001	1669926377	1669933577		
mega_sales_tshirts	002	1669926377			1669943577
abandoned_carts	003	1669915577		1669940777	1669944777
abandoned_carts	004	1669915577	1669941777		

With this attribution table, you can then infer how many of the business outcomes came as a result of which marketing message campaign. If you log additional information like purchase value, you can easily calculate the ROI of the campaign (total purchase value/ message send cost).

The benefit of logging timestamps is that you can configure the attribution window on the fly. This allows you to view the attributions based on a short window (e.g. 1 day) or a long window (e.g. 7 days). Longer attribution windows will allow more outcomes to be attributed to campaigns but may have less of a correlation (i.e. this purchase happened because of the marketing message). Longer attribution windows, however, may be more applicable for use cases that involve high-value products or services, since consumers typically need longer consideration time.

In-thread events



It’s important to understand the full customer journey — from the moment a customer enters a conversation to when the desired business outcome has been achieved. You can analyze where in the conversation customers drop off at so you can make necessary optimizations to the conversation flow. This way, you can capture every opportunity to its fullest and bring them down to the bottom of the funnel.

To answer questions like:

- Which product categories/items were users most interested in?
- What requests was your bot not able to handle that lead to handover to live agent?
- How long does payment take after an order has been created?
- Which payment method is more popular/faster?

You should track user events where possible, such as:

- When a conversation has started
- If you offer menu tree selections, which choice was selected
- When requested to speak with live agent
- When a sales order has been created
- What payment has been arranged

Where and how to track user events will depend on your implementation, but the idea is the same: the moment an event you’re interested in is detected by your implementation logic, you want to log that event immediately.

Bot-based

If the conversational chatbot is designed in a flow-based manner, you can simply classify events based on the state of the conversation. For example, if the user clicks on a “Show Details” button in chat, you can log an event “Show Product” to indicate that this user had some level of interest in the product.

NLP-based

If the conversational chatbot is implemented with natural language processing (NLP) to handle free-form input from the user, you can leverage other techniques to more granularly determine in-thread events.

The most basic but often powerful enough technique is word matching. To do this, first you need to come up with a mapping between words and meaningful events, such as account creation, payment made, shipment address entered. The word matching can happen in real time when the message is received or can happen later if all messages are logged in your database. It's recommended that you keep a good amount of conversations logs, and manually or semi-automatically mark the conversations logs in the event, in order to test different mapping logics and keep improving.

Off-thread events

If the checkout flow happens on the web and thus out of the WhatsApp thread, you can create links with referral parameters in order to track conversions that were generated from WhatsApp. For example, when you send a checkout link to a customer, you can append conversation-related data, such as phone number (hashed), message template name, uniquely generated codes, etc.

For example:
https://your websitehere.com/buy?productId=1234&**phone=hashed_phone_number&source=mega_sales_tshirts**

When your website loads this URL, it can parse the conversation data and store it in backend systems for measurement. When the user makes a purchase via this link, you can now attribute this purchase using the conversation data.

The same can apply for app deeplinks.

Analytics and reporting

Building the dashboard

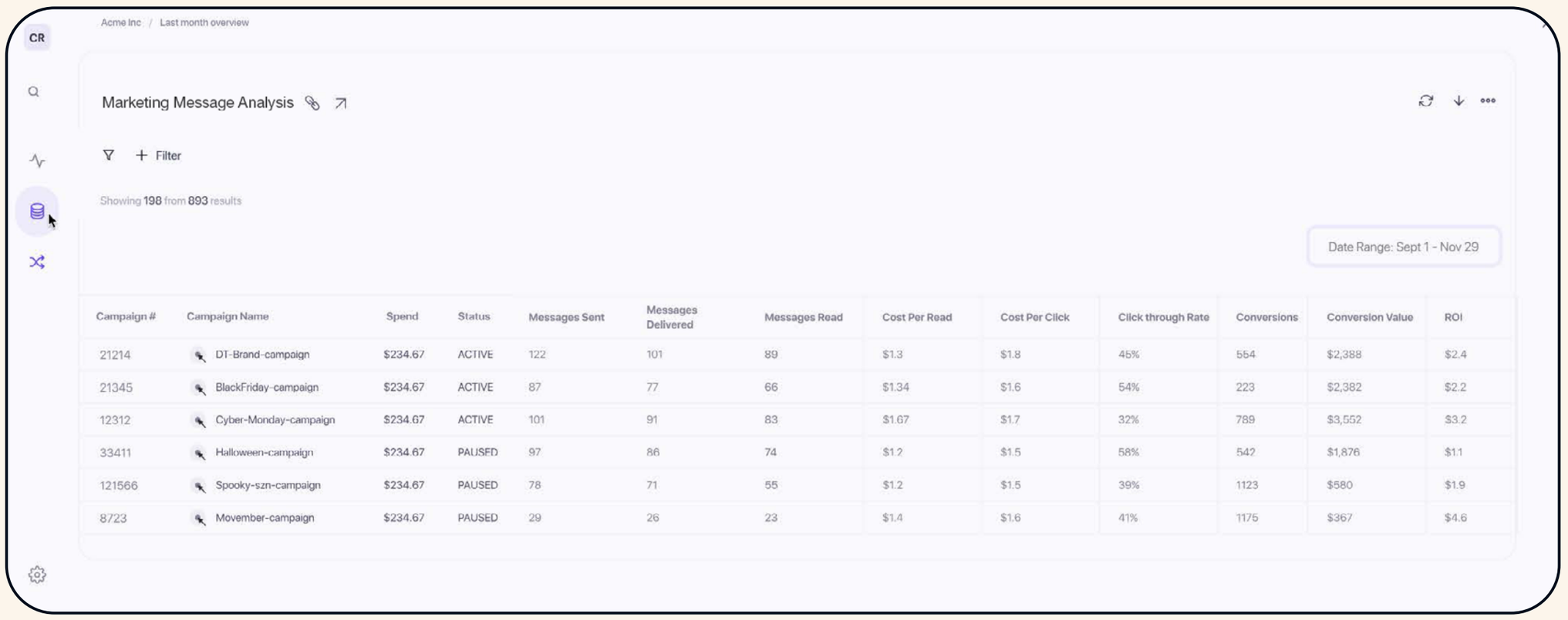
The dashboard is where we demonstrate the marketing messages outcomes. Typically the following metrics are of interest:

1. Total spend on marketing messages for a particular campaign
2. Business outcomes from the marketing message, which includes count of message reads, chatting thread, conversions, purchases, subscriptions, amount of conversions/purchases, etc.
3. Efficiency in getting the outcomes, which can be derived by dividing the outcome metrics by the cost metrics

On a higher level, providing aggregated results can also be helpful for strategic decisions on budget allocation across channels. Most of the channel-level measurements about the direct outcomes can be derived by correct aggregation of campaign-level measurements. Meanwhile, some indirect outcome measurements, which sometimes require more integration with a CRM tool, would also provide meaningful information (ex. reoccurring engagement with the WhatsApp account, repeated purchase, count and amount of upselling, etc.).

Example metrics to surface in the dashboard:

- Marketing messages campaigns
- Status of campaign
- Amount spent
- Messages sent
- Messages delivered
- Messages read
- Cost per read
- Cost per click
- Click-through rate
- Conversions
- Conversion value
- ROI



Campaign #	Campaign Name	Spend	Status	Messages Sent	Messages Delivered	Messages Read	Cost Per Read	Cost Per Click	Click through Rate	Conversions	Conversion Value	ROI
21214	DT-Brand-campaign	$234.67	ACTIVE	122	101	89	$1.3	$1.8	45%	554	$2,388	$2.4
21345	BlackFriday-campaign	$234.67	ACTIVE	87	77	66	$1.34	$1.6	54%	223	$2,382	$2.2
12312	Cyber-Monday-campaign	$234.67	ACTIVE	101	91	83	$1.67	$1.7	32%	789	$3,552	$3.2
33411	Halloween-campaign	$234.67	PAUSED	97	86	74	$1.2	$1.5	58%	542	$1,876	$1.1
121566	Spooky-szn-campaign	$234.67	PAUSED	78	71	55	$1.2	$1.5	39%	1123	$580	$1.9
8723	Movember-campaign	$234.67	PAUSED	29	26	23	$1.4	$1.6	41%	1175	$367	$4.6

Other benchmarks

If other channels are being used for marketing messages (SMS/email/in-app push notifications), it would be good to be able to pull ROI from these channels as well to understand how WhatsApp stacks against these channels.

Tools for logging

There are multiple ways of logging the incoming events for reporting purposes. It could be using an in-house solution or using third-party analytics tools like Google Analytics, Mixpanel, Heap Analytics, Plausible Analytics (open source), umami (open source), Fathom, Simple Analytics, Cube and others.

Event tracking leverages a custom code snippet that you add to the elements that you want to track on your website or app. Whenever users interact with that element, the code tells the third-party analytics tool to record the event.

Here is some guidance on how to set up tracking with a few widely popular analytics tools.

Using third-party analytics tools

Popular third-party analytics tools and platforms provide easy-to-use SDKs to log and pull data (writes and reads). Under the hood, these make an API call to the platform to record events. If you are familiar with or have used the Messenger Events API, then integrating with any third-party analytics tool should be fairly simple. You can refer to their developer docs on how to link your site to the analytics tool, configure the events needed for the use case, add event tracking to your website and find the event report.

At a high level, it would be something like this: POST /<api-endpoint> with parameters such as event_name, event_type, timestamp and other additional metadata.

```
Unset
// Example demo api call
curl \
  -X POST \
-H "Content-Type: application/json" \
-d '{
  "identity": "bob@example.com",
  "event": "add_to_cart",
  "timestamp": "2017-03-10T22:21:56+00:00",
  "metadata": {
    "item": "A90110",
    "quantity": "2",
    "price": 89.10,
    "currency": "USD",
  }
}' \
/<API-ENDPOINT>/log
```

Chapter 2: A/B testing to assess effectiveness of WhatsApp marketing messages

This next chapter provides best practices on how to measure WhatsApp marketing messages effectively, understand how many incremental conversions businesses can get as well as how to compare the effectiveness of marketing messages against other external platforms such as email/SMS. A brief summary of how these tests are performed is below:

Summarized testing steps

1. Pick a business use case to test. For example, a finance institution could create a campaign to offer a new credit card for current customers. The success metric here would be credit card signups from current customers.

2. Define if the WhatsApp marketing messages test will also contain benchmarks against other external platforms such as email or SMS. The experimental design will be based on how many benchmarks will be tested. The treatment groups will be:

    a. Control: customers that will not receive any message

    b. WhatsApp marketing message: customers that will receive the marketing communication

    c. Other cells for additional channel(s) comparison

3. Select metrics that will be important for assessing the effectiveness of marketing messages. Ultimately, the most important metrics are incremental conversions and cost per incremental conversions, but other upper funnel metrics can be tracked.

4. Use audience sizes and conversions for previous WhatsApp marketing messages campaigns and perform a power calculation to be able to detect at least a 10% increase in conversions between control and treated groups. Different percentage increases can be defined if appropriate.

    a. When comparing WhatsApp marketing messages vs. a holdout (Type A analysis), the power calculation can focus on detecting changes on conversion rates only.

    b. When comparing between external platforms/benchmarks (Type B analysis), the power calculation analysis should focus on improvements on cost per action (Conversion, CPA).

5. Execute the test, monitoring messages delivered in each treatment cell to calculate costs appropriately.
6. Track conversions from each treatment group.
7. Calculate test results metrics:
    a. Number of conversions between control and treatment groups
    b. Cost per incremental conversions, if the test is against holdout only
    c. Average improvement in CPA (cost per action) if comparing WhatsApp marketing messages with external platforms (email/SMS)
8. Discuss if final results are good enough to incentivize broad adoption of WhatsApp marketing messages or if any experiment iteration is necessary (test new message, new frequency, new benchmark channels, etc).

Research details

Business use case and measurement capabilities

Start by picking a business use case of WhatsApp marketing messages that have high potential to be successful. This can be assessed through previous marketing message campaigns that were executed where products have high business value and relevant conversion rates. This can be selected based on business sense and doesn’t necessarily need to have high precision, as the real result will be assessed by the experiment, but should somewhat reflect the reality.

The outcome selected needs to be transactional and measured, because we will need to quantify how many transactions happened to calculate the return on investment. For example, a financial institution wants to upsell a credit card to current customers and the transaction event is the credit card signup. Another example would be an ecommerce website wants to upsell a product and the transaction event is the purchase.

Measuring those outcomes needs to be possible at the end user or customer level. The bare minimum requirement is measurement at phone number level, but if comparing to an email marketing channel, each phone number needs to be attached to an email so we can measure all channels appropriately.

Research design

Define if you want to benchmark WhatsApp marketing messages against other scaled channels such as email marketing and SMS. We strongly recommend this comparison to help us understand how valuable marketing messages are across channels. The target list of customers should have the same probability of assignment in each of the cells tested (we call this process randomization) and, ideally, should be split equally. The easiest way of doing this is assigning them randomly in the following example cells:

(Experiment Type A) Incrementality of WhatsApp marketing messages

50%	50%
Control	WhatsApp marketing messages

(Experiment Type B) Benchmarks

50%	50%
SMS or email marketing	WhatsApp marketing messages

The percentages above are referring to how much of the audience or budget should be split between each treatment group (cell). We need to make sure that we have sufficient audience to detect significant differences — those differences need to be not only business relevant, but also statistically significant. To achieve this, we need to perform power calculations. If you want to benchmark against multiple channels, we recommend doing it at the same time, for example: WhatsApp marketing message (33%), SMS (33%) and email (33%).

Power calculation

We perform the power calculation analysis to make sure that the experiment has a sufficient audience size that will allow us to detect differences. An experiment that is not well-powered will likely generate false negative results for the effect size that we are looking for. For example, if we believe that a 10% difference in cost per action (transaction/ purchase) is important enough, our experiment needs to have a sufficient audience size for it. Otherwise, the experiment could only detect differences that are bigger than 10%.

According to the research design, we have two paths for power calculation:

Incrementality of WhatsApp marketing messages (Type A)

This power calculation is straightforward as it requires a minimum difference for effect size between two proportions. In this research type, the cost is only applied to the treatment cell, so we can calculate the cost per incremental conversion metric. We only need to power this experiment to be able to detect a minimum difference between conversion rates, at 80% power, 5% significance level and a minimum lift of 10%. The following code reproduces an hypothetical scenario and how power analysis is executed on R:

```
# Power Analysis for Incrementality:
control_cvr = 0.05 # Benchmark from previous WhatsApp Campaigns. Conversion Rate = 5%.
mde = 0.1 # Minimum effect size

# Power calculation for proportions
power.prop.test(
   p1 = control_cvr,
   p2 = control_cvr * (1 + mde),
   sig.level = 0.05,
   power = 0.8,
   alternative = "two.sided"
)

-- Output:

    Two-sample comparison of proportions power calculation

              n = 31233.44
             p1 = 0.05
             p2 = 0.055
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

According to the example above, we need to execute an experiment where we will be sending messages to 31,234 people in the control group (they don’t receive any WhatsApp marketing messages) and another group with 31,234 people that will receive WhatsApp marketing messages (test group). Important: The assignment of each person into each group (test or control) needs to be randomized.

Benchmark of WhatsApp marketing messages against other channels (Type B)

For this research design, the power analysis is more complex because the difference in conversion rates doesn't take into account the difference in cost between the alternatives. In this case, we need to estimate a baseline of conversion rates for the control group that equalizes for the difference in costs. Doing that, we will have a well-powered experiment to be able to detect a specific difference in cost per action (purchases).Take the following steps as an example:

1. Let's say that we want to benchmark WhatsApp vs. SMS, and the cost to send a message in each channel is USD 0.05 and USD 0.02 respectively. Also, from historical data, we know that the conversion rate for SMS is 1.5%.

2. Then we calculate the cost per message sent ratio between WhatsApp and SMS: 0.05 / 0.02 = 2.5.

3. The next step is to use this multiplier times the conversion rate of SMS to define a new baseline: 2.5 * 0.015 = 0.0375.

4. Then we proceed with the power calculation method for two different proportions. The baseline for control will be the number calculated in the step above (0.0375) and the test cell will be the baseline * the minimum expected lift (let's pick 10% improvement):

    a. Baseline (control) cell: 0.0375

    b. Test cell: 0.0375 * 1.10 (10% improvement) = 0.04125.

The following code reproduces a hypothetical scenario and how the power analysis is executed on R:

```
sms_cost = 0.02
wa_cost = 0.05
sms_cvr = 0.015
wa_cvr = (wa_cost/sms_cost)*sms_cvr
min_conversions = 75 # Minimum amount of conversions
sms_n = min_conversions / sms_cvr # Minimum sample size for SMS for power calc.
wa_n = min_conversions / wa_cvr # Minimum sample size for WA for power calc

library(pwr)

pwr.2p2n.test(
   h = ES.h(p1 = wa_cvr, p2 = wa_cvr * (1 + mde)), # Note that we have baseline CVR here vs
baseline * 10%
   n1 = sms_n,
   n2 = wa_n,
   sig.level = 0.05,
   alternative = "two.side"
)

-- Output:
    difference of proportion power calculation for binomial distribution (arcsine
transformation)

              h = 0.01928697
             n1 = 5000
             n2 = 2000
      sig.level = 0.05
          power = 0.1127482
    alternative = two.sided

NOTE: different sample sizes
```

Note that in the example above, we have to estimate initial sample sizes for each group (n1 = 5000, n2 = 2000). This was calculated based on the minimum amount of impressions needed given the conversion rates in each group to get at least 75 conversions. This number 75 is a result of previous experiences at Meta on minimum requirements for power calculation. The estimated power for this given scenario is 11.3%, which is insufficient (we should target 80%). So, the recommendation here is to increase the minimum sample sizes (n1 and n2) at the same rates until we get power = 0.80. After some trial and error, we were able to find that 15X more audience will give us 80% power:

```
pwr.2p2n.test(
   h = ES.h(p1 = wa_cvr, p2 = wa_cvr * (1 + mde)), # Note that we have baseline CVR here vs
baseline * 10%
   n1 = sms_n * 15, # 15x audience multiplier
   n2 = wa_n * 15, # 15x audience multiplier
   sig.level = 0.05,
   alternative = "two.side"
)
-- Output:
    difference of proportion power calculation for binomial distribution (arcsine
transformation)

              h = 0.01928697
             n1 = 75000
             n2 = 30000
      sig.level = 0.05
          power = 0.806031
    alternative = two.sided

NOTE: different sample sizes
```

Now, to be able to detect a 10% improvement in cost per action (conversion), we need to have the following sample sizes:

- SMS treatment: 75,000 people reached
- WhatsApp treatment: 30,000 people reached

Once the audience requirements are defined, clients will proceed with randomization of the audience in each treatment group and execution of the test. Before executing the test, it’s important to check if randomization was executed appropriately.

Randomization

In this step, you will divide your customers into the treatment groups. If you're running an incrementality analysis (type A), you will have the control (holdout) group and the test group (will receive WhatsApp messages). If you're running the benchmark analysis (type B), you will have to randomize the audience in the same way, but in the respective treatment groups.

The assignment should be done randomly to guarantee that the only aspect that differs between the groups is the treatment. To test if your randomization was executed well, you can perform some group comparison tests for several metrics that can define historical behavior or demographics, such as:

- Number of WhatsApp messages received in the last 28 days
- Number of SMS messages received in the last 28 days
- Number of purchases in the last 28 days
- Age, gender, location, etc.

You should evaluate if those metrics differ, on average, between groups before executing the test. This can be done using a t-test for continuous variables or chi-square tests for discrete variables. The following example presents some simulated data for number of WhatsApp messages received in 28 days, other continuous variables, some descriptive statistics and t-tests comparing control and test groups using the package tableone on R:

```r
library(tableone) # This package helps scaling t-tests for all metrics that you have in
your table.

table1 = CreateTableOne(
    vars = c("wa_messages_28d", "email_messages_28d", "purchases_28d"),
    strata = "group",
    data = base_data,
    test = T
)
summary(table1)

-- Output:

### Summary of continuous variables ###

group: control
                       n miss p.miss mean sd median p25 p75 min max skew kurt
wa_messages_28d    10000    0      0    5  2      5   3   6   0  15  0.4  0.1
email_messages_28d 10000    0      0   10  3     10   8  12   1  23  0.3  0.1
purchases_28d      10000    0      0    3  1      3   2   4   0   9  0.3 -0.2
------------------------------------------------------------
group: test
                       n miss p.miss mean  sd median p25 p75 min max skew kurt
wa_messages_28d    10000    0      0    5 2.2      5   3   6   0  14  0.4 0.02
email_messages_28d 10000    0      0   10 3.1     10   8  12   1  25  0.3 0.17
purchases_28d      10000    0      0    9 0.9      9   8  10   5  10 -0.8 0.40

p-values
                     pNormal pNonNormal
wa_messages_28d    0.1639733  0.1593824
email_messages_28d 0.7323478  0.6499114
purchases_28d      0.0000000  0.0000000

Standardize mean differences
                        1 vs 2
wa_messages_28d    0.019684200
email_messages_28d 0.004836736
purchases_28d      4.882094580
```

For the list of variables, you should expect that none of them differ statistically significantly at 5%. Interpreting the results:

- In the example above, you can see in the p-values section that the groups don't differ for wa_messages and email_messages, because p-value is greater than 0.05. This is what we're looking for.
- However, for purchases_28d, we can see that p-value is smaller than 5%, indicating that they differ. You can see in the first two tables that average purchases in the control group is 3 and in the test group is 9, which is very different.
- In this case, it looks like the groups were not very well randomized, so we recommend to run randomization again and perform the test again, until all variables show p-values > 0.05.

Once you get balanced groups with p-values > 0.05 for all variables, you can move forward with the data analysis step.

Data analysis

The data analysis is executed depending on which type of research design was selected. In summary, each analysis type has a different focus on what metric is compared against control (holdout) and test groups:

1. **Incrementality of marketing messages (Type A)**: for this type of analysis, the focus is on comparing the conversion rates between holdout and test group using a proportions test. The outcome of this analysis is the conversion rates (CVR) confidence intervals of each group as well as a p-value for the difference in proportions. The test is executed through an analytical solution and is presented below with an example.

2. **Benchmark of marketing messages against other channels (Type B):** for this type of analysis, the focus is comparing the cost per action (CPA), where the action is the conversion event, between channels. The outcome of this analysis is the CPA confidence intervals of each group as well as a p-value for the difference in CPAs. The test is executed through simulations given the experimental data with the following steps:

   a. Simulate 10,000 conversions for each treatment group using a binomial distribution where parameters that are included in the simulation function are number of messages sent and conversion rate (conversions/messages sent).

   b. Calculate cost per action CPA (conversion) for each simulation.

   c. Take the confidence intervals for CPAs using percentiles 2.5% and 97.5% in each treatment group.

   d. Calculate delta between CPAs of each group for each simulation.

   e. Calculate the p-value using the proportion of cases where delta > 0.

The sections below provide examples on how to perform the data analysis in R.

Data analysis for incrementality of marketing messages (Type A)

For this example, let's assume that we used the power calculation example above and we landed an experiment with 32K customers in both holdout/test groups. The holdout group, where the message was not sent, had 1,430 conversions, and the test group, where the message was sent, had 1,705 conversions. Assuming the cost of USD 0.05 per message sent, we have the following code:

```
# Assuming we run the experiment with the proper power mentioned earlier:
n = 32000 # Sample size in each group
holdout_conversions = 1430 # Provided by client/partner
test_conversions = 1705 # Provided by client/partner
wa_cost = 0.05

# Test for difference in proportions
diff = prop.test(
   x = c(holdout_conversions, test_conversions),
   n = c(n, n)
)

holdout_ptest = prop.test(holdout_conversions, n)
wa_ptest = prop.test(test_conversions, n)

# Showing results from the tests/confidence intervals
diff
holdout_ptest
wa_ptest

# Calculating cost per incremental conversions
incremental_conversions = test_conversions - holdout_conversions
cost_per_incremental_conversions = (wa_cost * n) / incremental_conversions
Cost_per_incremental_conversions

-- Output:

        2-sample test for equality of proportions with continuity correction

data:  c(holdout_conversions, test_conversions) out of c(n, n)
X-squared = 25.181, df = 1, p-value = 5.219e-07
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.011968679 -0.005218821
sample estimates:
    prop 1     prop 2
0.04468750 0.05328125

        1-sample proportions test with continuity correction

data:  holdout_conversions out of n, null probability 0.5
X-squared = 26534, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.04246258 0.04702248
sample estimates:
       p
0.0446875

        1-sample proportions test with continuity correction

data:  test_conversions out of n, null probability 0.5
X-squared = 25542, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.05085838 0.05581204
sample estimates:
        p
0.05328125
```

So, here is a summary of important insights:

- Average of conversion rates:
    - Holdout: 0.0447.
    - Test: 0.0533.
    - Lift: 19.23%.
- Confidence intervals for conversion rates:
    - Holdout: [0.0424, 0.0470].
    - Test: [0.0508, 0.0558].
    - Difference between groups: [-0.0120, -0.0052].
- P-value for the difference between conversion rates: < 0.001.

Putting the results in a summary table and charts we have:

group	cvr	lwr	upr	message_sent	cost	conversions	p.value	incremental_conversions	cpic
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
holdout	0.04468750	0.04246258	0.04702248	NA	NA	1430	NA	NA	NA
whatsapp	0.05328125	0.05085838	0.05581204	32000	0.05	1705	1e-06	275	5.82

This test showed a statistically significant difference between conversion rates of holdout and test group (p-value < 0.05). This is evidence of incrementality of WhatsApp marketing messages and the cost per incremental conversion (CPIC) was $5.82, with an estimated lift of 19.23%.

Data analysis for benchmark of marketing messages with other channels (Type B)

For the benchmark data analysis example, let's assume the power calculation above where we sent messages to 75K people through SMS and 30K people through WhatsApp, so we would have enough power according to the conversion rates presented for that power calculation. After delivering the experiment and tracking the results, we had the following data:

Metric	SMS	WhatsApp
Messages sent	75,000	35,000
Cost per message sent	0.02	0.05
Conversions	1,490	1,678

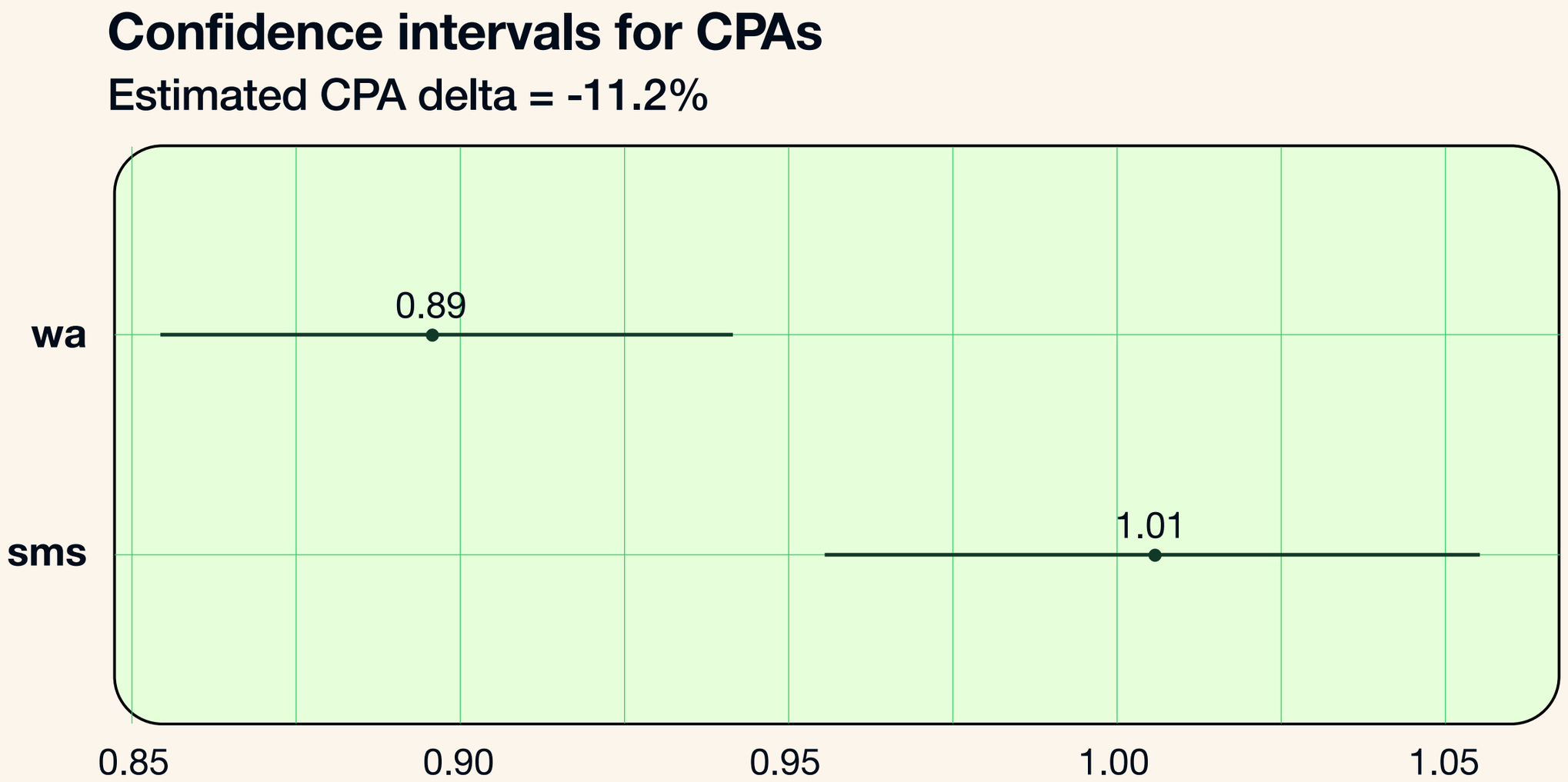
The code below will run the simulation with 10,000 random conversions given experimental data and calculate results:

```
# Setting parameters
sms_n = 75000 # SMS messages sent
wa_n = 30000 # WhatsApp messages sent
sms_conversions = 1490 # Conversions from SMS group
wa_conversions = 1678 # Conversions from WhatsApp group
sms_cost = 0.02 # Cost per message sent on SMS
wa_cost = 0.05 # Cost per message sent on WhatsApp

# Simulating conversions given experimental data
sms_sim = rbinom(10000, sms_n, sms_conversions/sms_n)
wa_sim = rbinom(10000, wa_n, wa_conversions/wa_n)

# Calculating CPAs per treatment group
sms_sim_cpa = (sms_n*sms_cost)/sms_sim
wa_sim_cpa = (wa_n*wa_cost)/wa_sim

# Confidence Intervals for Each Treatment
summaryTable = data.frame(
   group = "sms",
   avg_cpa = mean(sms_sim_cpa),
   lwr = quantile(sms_sim_cpa, 0.025),
   upr = quantile(sms_sim_cpa, 0.975)
) %>%
   bind_rows(
       data.frame(
           group = "wa",
           avg_cpa = mean(wa_sim_cpa),
           lwr = quantile(wa_sim_cpa, 0.025),
           upr = quantile(wa_sim_cpa, 0.975)
       )
   )

rownames(summaryTable) = c()
summaryTable

# Confidence Intervals for Delta
delta = wa_sim_cpa/sms_sim_cpa - 1

deltaSummary = data.frame(
   avg_delta = mean(delta),
   p.value = mean(delta > 0),
   lwr = quantile(delta, 0.025),
   upr = quantile(delta, 0.975)
)
rownames(deltaSummary) = c()
deltaSummary
```

group	avg_cpa	lwr	upr
<chr>	<dbl>	<dbl>	<dbl>
sms	1.0073693	0.9584665	1.058574
wa	0.8942607	0.8537279	0.937500

avg_delta	p.value	lwr	upr
<dbl>	<dbl>	<dbl>	<dbl>
-0.1117189	4e-04	-0.171331	-0.05047879

These two tables summarize important findings, where the delta of CPAs between WhatsApp and SMS is -11.2%, showing that each conversion obtained using WhatsApp cost 11.2% less than each conversion obtained using SMS. The confidence intervals for delta don't contain zero, and p-value is <0.05, indicating that this difference is statistically significant. Using charting, you can plot the confidence intervals and averages for proper visualization:



These results indicate that WhatsApp is more cost-effective than using SMS for the data presented above.

Final considerations

The measurement playbook above provides recommendations on best practices on how to measure WhatsApp marketing messages campaigns with two approaches: incrementality and channel benchmarking. These use frequentist methods and simulation according to the metrics analyzed. We understand that running statistical tests and simulations is not trivial, but it’s important to follow each step of this process, so we can maintain the integrity of the data generated with the experiments.

from

Meta