

Preamble on different approach & scope of service	1
Scope of service	2
Source and methodology for estimating the universe	3
Panel recruitment, demographics, size, weighting, health and approach to privacy	5
Panel Recruitment	5
Panel demographics	7
Panel size	9
Panel weighting	9
Panel health	11
Approach to privacy	12
Overall description of how AudienceProject measures	12
Estimation of online reach	16
The simple model explained	16
The complex model explained	18
Getting Frequency right	22
Key challenges to frequency measurement	22
Challenge 1: Cookie-death	23
Challenge 2: Ad-blockers / cookie-rejects	25
Challenge 3: First party versus third party	26
What can be done?	27
Research panels	27
AudienceProjects big data approach	27
Humans. Not devices.	28
The deterministic approach	28
The probabilistic approach	29
AudienceProject's approach to cross device	29
Appendix A - Detailed description of sources for universes	32
Appendix B – Content of columns in W3C extended log files	33
Appendix C - Identification of non-human traffic	40
Appendix D - Overview of identifier lifecycle	42
Cookie lifecycle	42
Creation	42
Activation and contribution to measurement	42
Discontinuation	43
Example	43
Mobile Identifier lifecycle	44
Creation	44
Activation and contribution to measurement	44

Discontinuation	44
Example	45

Preamble on different approach & scope of service

In a world where people use an ever-increasing number of devices, the ability to get the frequency right is central to Audience Measurement. Therefore, quite understandably, we often receive questions on the nature of the cross device measurement system powering the frequency counts which are key to a lot of the metrics in our audience measurement service AudienceReport.

As our service relies on rather complicated technology, statistics and data, we will try to outline in as non technical terms as possible first why the problem is indeed a hard one, then different approaches to deal with the issue and, finally, our approach to deliver precise measurements.

For starters, we offer some reflections as to why the issue has not so far been addressed particularly well by anyone active in the field. We believe it has to do with the fact that it is inherently hard to marry approaches originating from separate industries with different worldviews and skill sets.

Different industries. Different approaches.

Quite tellingly, we tend to receive very different types of questions depending on which venue of business the client comes from. From market research people, often with a history of Television Audience Measurement or classical market research, we tend to receive questions around our panels, stratification procedures i.e. From digital people as well as software engineers, we receive questions relating to how fast our systems work, how we integrate with others, how our tracking pixels are deployed etc.

On the face of it, this may appear somewhat counterintuitive. Why should our service generate questions from two rather diverse groups of users?

But it is in fact quite understandable since AudienceProject and therefore our service AudienceReport in a very fundamental way is a hybrid between market research and software engineering in the media space (and other areas as well).

Breaking down previous industry boundaries

Often, truly transformative services arise at the intersection of different industries, in turn leading to a breakdown of previous industry boundaries. This is indeed the case for the measurement service AudienceReport (and for our entire company) as it sits between software engineering and market research, dissolving the barriers between the two.

This leaves many people coming from, perhaps especially, the market research space with unanswered questions.

In the new world, you buy audiences as well as media

In the market research industry this dynamic is unfolding in a somewhat similar fashion, as at least until very recently, few incumbents understood that technology would lead to a very fundamental change in the way ad campaigns will be measured and in the way media-planning will be tied to the actual real-time execution of campaigns through means of delivery of data-based real time audience targeting. In the new world, you don't just buy media. You buy audiences and you want to make sure that the audience is built, validated and executed upon in a seamless fashion across its multiple devices.

Increasingly, this change puts the existing Television Audience Measurement (TAM) panels consisting of app. 1.000 households under pressure when it comes to measuring ads delivered digitally (WebTV etc.). Perhaps most telling in this respect, as TV-publishers connect more and more of their inventory

to programmatic platforms, the inability of classical TAM systems to validate a (for this industry) foreign concept such as a third party dataset applied on a TV-stations programmatic inventory becomes evident for all to see.

Re-thinking the concept of panels

Much of the thinking in this area originates from the old TAM world, where only a few big TV-stations existed and very few 'devices' were available for watching TV. In that world, it made sense to operate with small panels of viewers and then focus all available resources on persuading those viewers to turn on their specific remote-control when watching TV. Collecting vast amounts of survey-based background information for each participating viewer would also be necessary in order to ensure proper demographic categorisation.

But is this an efficient use of resources in a digital world characterized by:

- **An extremely long tail of media** in which the campaigns are executed. Often, a campaign will run across 100+ media as opposed to a few big TV-stations. Hence, larger panels are inherently needed to be able to plan meaningfully across medias.
- **An inherent inability** to ensure real-life compliance by panelists on all their devices. Regardless of how much you incentivize and induce you cannot make sure that your panelists register viewing on all their 8 or more devices (source: [Latest AudienceProject device-survey](#)).
- **Multi-device usage** by most people. Is a narrow panel concept even meaningful when measuring the true frequency of a campaign, the normal key argument behind small panels?

In our experience, it is very important to understand that the challenges related to proper audience measurement in the digital space can only be overcome, if you intelligently couple new technologies with panels that are much bigger than classical TV panels, yet recruited diversely and weighted and managed properly and fused with information that are not only survey-based.

Scope of service

One of AudienceProject's key take-aways from this is that measuring frequency by recruiting a small panel of respondents that are required to install software trackers on all of their devices is not sustainable. It must be some form of passive measurement that becomes the solution in order to accurately measure long tail traffic.

Therefore, AudienceProject's panel doesn't rely on a VPN or "router-meter", but measure using the identifiers emails, device-ids, cookies and, in the absence of any of those identifiers (fx for addressable TV through fx Apple-TV or Roku), specific combinations of IP-addresses and user-agent strings.

The identifiers are captured passively either as the panelist signs up to our panel or assigned using our knowledge graph - See section on panel recruitment and getting frequency right for details.

The logfiles/census data is captured by inserting a simple 1x1 pixel into either the creative (for ads) or the header or footer of the site (for site centric measurement).

This means that as long as the pixel is able to fire, AudienceProject is able to measure - There is therefore no limitation for device (in-app is not problem), operating systems (IOS is not a problem) nor browsers (Safari is not a problem).

Source and methodology for estimating the universe

Our relevant universes and source materials are as follows.

Full population metrics are derived from Statistics Finland population metrics.

Population

Finnish population numbers from Statistics Finland

Age	Absolute distribution			Relative distribution		
	Men	Women	Total	Men	Women	Total
16-24	304.308	291.583	595.891	8%	7%	15%
25-34	354.975	335.739	690.714	9%	8%	17%
35-44	335.979	319.987	655.966	8%	8%	16%
45-54	374.077	369.121	743.198	9%	9%	18%
55-64	371.929	383.478	755.407	9%	10%	19%
65-74	278.911	313.240	592.151	7%	8%	15%
Total	2.020.179	2.013.148	4.033.327	50%	50%	100%

In order to estimate the online population used to weight our panel and define the Finnish online universe, we rely on yearly “Use of information and communications technology by individuals” produced by Statistics Finland on a yearly basis (see full description in appendix A).

Online penetration

Online penetration rates from Statistics Finland applied to the Finnish population

Age	Online penetration			Absolute distribution		
	Men	Women	Total	Men	Women	Total
16-24	99%	99%	99%	301.265	288.667	589.932
25-34	100%	100%	100%	354.975	335.739	690.714
35-44	100%	100%	100%	335.979	319.987	655.966
45-54	96%	96%	96%	359.114	354.356	713.470
55-64	90%	90%	90%	334.736	345.130	679.866
65-74	68%	68%	68%	189.659	213.003	402.663
Total	92%	92%	92%	1.875.728	1.856.883	3.732.611

The population is broken down into a gender-age matrix, where the cell-count is adjusted according to the information from “Use of information and communications technology by individuals” about the share of population within each cell that is actually active online. It's not enough to have to opportunity to go online, our weights and universes are based on the proportion that actually goes online according to the study.

Online population

Finnish online population defined by Statistics Finland

Age	Absolute distribution			Relative distribution		
	Men	Women	Total	Men	Women	Total
16-24	301.265	288.667	589.932	8%	8%	16%
25-34	354.975	335.739	690.714	10%	9%	19%
35-44	335.979	319.987	655.966	9%	9%	18%
45-54	359.114	354.356	713.470	10%	9%	19%
55-64	334.736	345.130	679.866	9%	9%	18%
65-74	189.659	213.003	402.663	5%	6%	11%
Total	1.875.728	1.856.883	3.732.611	50%	50%	100%

Panel recruitment, demographics, size, weighting, health and approach to privacy

Panel Recruitment

AudienceProject offers a free online survey that companies, organizations and other website holders can set up to evaluate their users' satisfaction with their website.

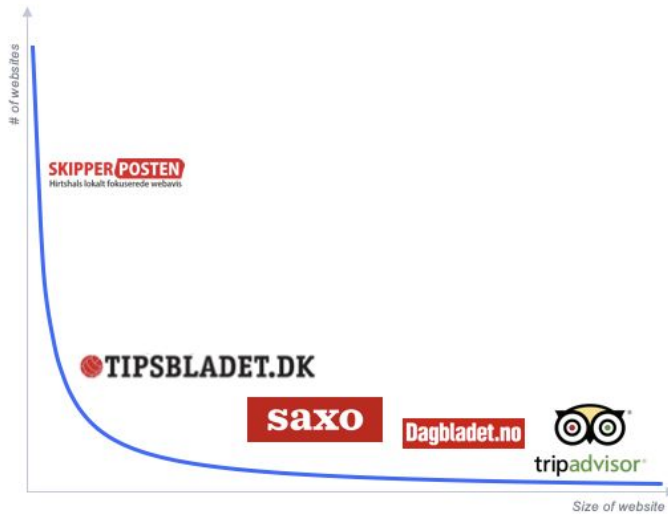
At the end of the survey, the respondent is asked, if they want to become a member of AudienceProject online survey panel.

You can take a test survey at this link: https://www.userreport.com/#_urp=test_invite

Please note that invitation overlay ensures that AudienceProject has first-party interaction with all our panelists.

This recruitment method has several advantages in creating a high level of representativeness in the panel and consequently in the surveys carried out in the panel.

WHAT KIND OF SITES USES USERREPORT



UserReport is used by both small local sites as well as big international corporations.

First, recruitment takes place on a large number of websites that are of different types - both the vast number of websites and the dispersed nature of them can be expected to affect representativeness in a positive way. Normally, the recruitment survey is active on more than 2.000 different websites at the time - historically recruitment to the panel has taken place on more than 18,000 different websites. Some panelists from preceding periods are still in the panel, so the variation in recruitment sources is vast.

DIVERSITY IN WEBSITES ENSURES DIVERSITY IN PANEL



Second, there is no option for self-invitation: people can only sign up to AudienceProject's panel by invitation from AudienceProject. Thereby, we avoid or minimize problems connected with having 'professional panelists' that can have a tendency to seek out and sign up to panels. The problem with professional panelists is in part that they can become familiar with questionnaires and survey method to an extent where it can influence their answers, and in part that they can be too focused on the rewards, which also can influence their answers.

In relation to this, we also only motivate through short text formulations combined with modest prize draws, so participation does not become about stockpiling points or payment - which can lead to data quality problems like speeding and incorrect answers. And it is important to note that members of online panel are not incentivised in any way; it is only, if a panelist is also a member of AudienceProject's research panel that they participate in the draw about gift card.

AudienceProject's online panel uses an intelligent targeting- and invitation systems that funnel respondents to create representativeness in the different surveys. The survey management system is designed to send new invitations to panelists that based on background variables are still in demand in the cells in all the active surveys (for example age/gender cells).

As a result the representativeness inside the conducted surveys can be approximated to the distributions of the general population. The distribution of our panelists on widely used background variables like age, gender, education, occupation and zip code are close to the distribution in the Finnish population.

Furthermore, representativeness and data quality is secured in the processes of data analysis - most notably by applying complex and meticulously constructed social science weighing methods. We will go into more details on this in below point around weighting.

Panel demographics

As part of the survey, the panelists are asked to 9 standard demographic questions - These questions are always part of the survey.

The answers to these 9 standard demographic questions form the basis for demographic variables in AudienceProjects online panel.

The standard demographic variables are:

- Country
- Zip code
- Gender
- Age
- Employment status
- Education level
- Household size
- Children in household
- Household income

In addition to the built in verification and error correction in the survey (fx check of valid zip code), there is a series of automatic checks of the answers post hoc, before a person is included in AudienceProject's online panel.

Which part of the country do you live in? Please enter your postcode:

23449

Please enter your zip code

OK Skip

In which country do you live?

You answered **Denmark**

Do you have any children in your household?

Yes	40%
No	60%

Skip

What is the highest educational level that you have attained?

Primary school	11%
Highschool	12%
Secondary school: technical/vocational type	5%
Secondary school: university-preparatory type	4%
University or college level education	68%

Skip

These automatic checks include, but are not limited to “speeders” or “skippers”, however, given the survey design they are not a major problem.

The automated check that blocks the most information/would be panelists from joining the panel is the automatic logical checks, where we look at the internal consistency of the answers provided - fx someone answering that they are 16 years old and living alone with two children will not be included in panel.

Panel size

Currently, the Audienceproject’s Finnish panel is around 76.000.

AudienceProject online panel

AudienceProject’s Finnish online panel split into Statistics Finland’s age and gender groups

Age	Absolute distribution			Relative distribution		
	Men	Women	Total	Men	Women	Total
16-24	1.993	5.122	7.116	3%	7%	9%
25-34	5.039	9.944	14.984	7%	13%	20%
35-44	6.658	9.411	16.070	9%	12%	21%
45-54	6.895	10.521	17.416	9%	14%	23%
55-64	5.701	7.796	13.497	7%	10%	18%
65-74	3.767	3.272	7.039	5%	4%	9%
Total	30.054	46.068	76.122	39%	61%	100%

Panel weighting

The population is broken down into a gender-age matrix, where the cell-count is adjusted according to the information from “Use of information and communications technology by individuals” about the share of population within each cell that is actually active online. It's not enough to have to opportunity to go online, our weights and universes are based on the proportion that actually goes online according to the study.

The resulting weight matrix is deployed as an iteratively estimated weight on the full panel. All weights and panels are estimated and assessed every 2 weeks. A total of 26 sub-panels each with a unique weight-profile is created per country.

AudienceProject online panel compared to online population

AudienceProject's Finnish online panel compared to the Finnish online population

Age	AudienceProject online panel			Online population			Weights applied		
	Men	Women	Total	Men	Women	Total	Men	Women	Total
16-24	3%	7%	9%	8%	8%	16%	3,06	1,19	1,71
25-34	7%	13%	20%	10%	9%	19%	1,51	0,69	0,97
35-44	9%	12%	21%	9%	9%	18%	1,03	0,73	0,85
45-54	9%	14%	23%	10%	9%	19%	1,10	0,67	0,83
55-64	7%	10%	18%	9%	9%	18%	1,20	0,88	1,02
65-74	5%	4%	9%	5%	6%	11%	1,01	1,40	1,19
Total	39%	61%	100%	50%	50%	100%	1,27	0,83	1,00

The weighting only affects the panel and the individual panelist's weight in the panel - No weighting is done to the census data (measured traffic).

Below we have added distributions for a selection of Finnish panels created in 2019. Additional panels can be exported upon request and similar metrics shared.

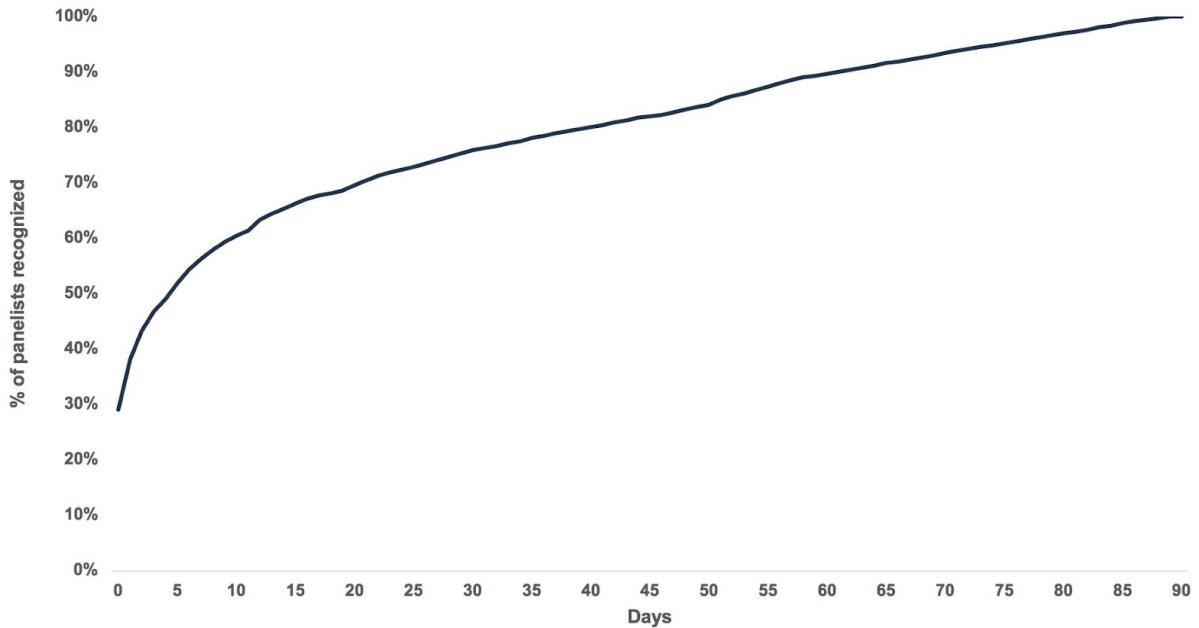
		17-06-2019	01-06-2019	16-05-2019	16-04-2019	15-03-2019
Mean		1.00000	1.00000	1.00000	1.00000	1.00000
95% Confidence Interval for Mean	Lower Bound	.99663	.99661	.99664	.99671	.99683
	Upper Bound	1.00337	1.00339	1.00336	1.00329	1.00317
5% Trimmed Mean		.95035	.94906	.94841	.94964	.95097
Median		1.00000	1.00000	1.00000	1.00000	1.00000
Variance		.15625	.15886	.15869	.15366	.14773
Std. Deviation		.39528	.39857	.39836	.39199	.38436
Minimum		.66789	.66761	.67220	.67815	.67865
Maximum		3.05660	3.06704	3.06124	2.97383	2.91457

Panel health

To qualify as a panelist, AudienceProject has to have seen the panelist within the last 90 days on one or more of the ad campaigns that we measure (or on one or more sites that we measure).

Recognition rate

Finnish panelists recognized within 90 days



AudienceProject sees around 70% of our panelists within a two week period, which is quite high, when you consider that AudienceProject currently mainly measures online ads & videos for the agencies in Finland.

Friendly reminder, AudienceProject doesn't install a VPN or a routermeter on our panelists, so we only see our panelists, when the are exposed to ad campaigns that we measure (or visit sites that we measure).

Another way to look at the health of our panel is to look at the panel stability. Starting point is panel at the beginning of March and looking forward 90 days WITHOUT including new panelists recruited in the period.

Finnish panel	1-03-2019	15-03-2019	1-04-2019	16-04-2019	1-05-2019	16-05-2019	1-06-2019
Panelists	75.217	71.832	69.103	65.371	62.625	59.870	57.355
Churn N (Opt-out & lost)	0	3.385	2.730	3.732	2.746	2.756	2.515
Churn %	0%	4,5%	3,8%	5,4%	4,2%	4,4%	4,2%
Survival %	100%	96%	92%	87%	83%	80%	76%

The net loss of panelists is only around 4% per fortnight; meaning that 76% of a panel panelists are still active after 90 days.

Approach to privacy

Privacy is built into everything Audienceproject do.

As is also evident from the link to test survey shared under the panel recruitment section; to join our panels, you either have to give explicit consent once (online panel) or twice (research panel).

As a panelist, you can easily opt-out of our panels and find detailed information about your data at:

<https://privacy.audienceproject.com/en-GB/for-users/opt-out>

You can read more about our approach to privacy at our privacy portal

<https://privacy.audienceproject.com/en-GB>

The links are to the English version, but you can (by using the dropdown menu to the right) easily find information in the local language of all the markets, where AudienceProject offer our services.

Overall description of how AudienceProject measures

AudienceProjects AudienceReport measures campaign performance by the application of a single 1x1 tracking pixel to each campaign or placement within a. When the pixel is fired, AudienceReport logs the request and process the logged information in order to build our reports.

It is important to note that AudienceProject have 3 different sources of information, when measuring a campaign or a site:

1. The logged entry itself
2. The pixel can contain parameters with embedded metrics (fx device ids)
3. Our proprietary panel of panelists

The pixel itself is pretty straightforward: <https://visitanalytics.userreport.com/hit.gif>. 1x1 SSL by default, but regular http can also be used. No cookies are dropped by these pixels – please remember that to the extent that AudienceReport relies on cookies for measurement, AudienceProject uses its own 1-party cookies.

In order to distinguish between different clients and different placements/dimensions of the campaign, each client will use unique t-codes embedded into the tracking-pixel:

[https://visitanalytics.userreport.com/hit.gif?t=\[t-code\]](https://visitanalytics.userreport.com/hit.gif?t=[t-code]) The purpose of the t-code is to classify dimensions as well as data ownership.

In addition to the t-code, an optional event parameter can also be added. If an event parameter is absent, AudienceReport will assume the event is on an impression level. A few examples of other events types are listed below:

A typical fully formed pixel could look like this example:

<https://visitanalytics.userreport.com/hit.gif?t=USRfd6cffb9&event=click>

Whenever a tracking pixel is fired by a client using AudienceReport to track a campaign an entry into our W3C extended log file format is created. Our tracking pixels are all hosted on AWS Cloudfront ensuring worldwide scalability and low latency.

Each log-file is streamed into our real-time log-processing architecture named AqueDuct. While we operate with an 5 minute SLA on individual log-files, we see much faster processing on average.

The W3C extended log file format contains the following columns (see appendix B for a description of the content of each columns of the log files):

date, time, x-edge-location, sc-bytes, c-ip, cs-method, cs (Host), cs-uri-stem, cs-status, cs (Referer), cs (User-Agent), cs-uri-query, cs (cookie), x-edge-result-type, x-edge-request-id, x-host-header, cs-protocol, cs-bytes, time-taken, x-forwarded-for, ssl-protocol, ssl-cipher, x-edge-reponse-result-type and cs-protocol-version

Individual events are essential to AudienceReports ability to measure campaigns and calculate reports on a granular level.

While most of our competitors rely on more traditional aggregated metrics and stale reports, AudienceReport works like Google Analytics (but without the sample size constraints). Metrics can be estimated and re-estimated on the fly. Data added and removed (by adding and removing tracking points/t-codes), target groups redefined, data-range altered and results filtered.

You will have been provided with access to our live reporting interface, we recommend playing around with it to get a better feel for the real-time & granular nature of the measurement with AudienceReport. The most critical components in the event logs (mentioned above and described in appendix B) are outlined below. If AudienceProject is to receive event-logs from a third party, the information derived from these columns will have to be provided in a comparable format. Since AudienceProject is a measurement and validation service, not a reporting service, AudienceProject need sufficient granularity of metrics to actually validate the delivery of each publisher, media, platform. Aggregated metrics that can't be validated independently are not of much use.

The most critical components are:

Date, time, IP, URL referrer, URL parameters, User-Agent, AP cookie (if present)

We will shortly walk you through each component:

Date: The date on which the event occurred in the format yyyy-mm-dd, for example, 2015-06-30. The date and time are in Coordinated Universal Time (UTC).

Time: The time when the CloudFront server finished responding to the request (in UTC), for example, 01:42:39.

Since events are reported on daily and hourly dimensions, the data time stamp is important in order to ensure proper processing of metrics.

The data-time stamp is also used to detect add collisions (same ad appearing multiple times to the same user on the same page-load).

IP: The IP address of the viewer that made the request, for example, 192.0.2.183 or 2001:0db8:85a3:0000:0000:8a2e:0370:7334. If the viewer used an HTTP proxy or a load balancer to send the request, the value of c-ip is the IP address of the proxy or load balancer.

IP address is used for multiple purposes.

Initially: Geo-location verification - is the impression delivered to a user in the intended country / geography? Since AudienceReport measures reach each impression needs to be matched to the correct country in order to derive reach in the population. We also verify that the delivery match the client's geographical requirements. Too often we see impression from third party networks originating from wrong locations.

Traffic quality evaluation. IP's are scored across a quality parameter. It allows us to evaluate how legit the incoming traffic is. Does it originate from well known bot-net IP's? Are we seeing attempts at artificially inflating numbers with bots? Proxy-servers etc. etc.

Referrer: The name of the domain that originated the request. Common referrers include search engines, other websites that link directly to your objects, and your own website + The query string portion of the URI, if any. When a URI doesn't contain a query string, the value of cs-uri-query is a hyphen (-).

The referrer is used to validate which medias the actual campaign is delivered from. Is the traffic indeed originating from the expected source? As part of our traffic quality estimate, attempts at obscuring referrers will result in low quality scores.

The referrer is also used for brand safety scoring. Does the impression execute in a brand safe environment?

User-Agent: The value of the User-Agent header in the request. The User-Agent header identifies the source of the request, such as the type of device and browser that submitted the request and, if the request came from a search engine, which search engine.

Used to identify and validate the device mix on ads served. Usually used for Desktop, Tablet, Mobile classification as well as Smart-TV and native app identification and classification.

URL parameters: The URL parameters included with the tracking-pixel request. Contains t-code, events and optional additional parameters like geographical coordinates (long/latt) and mobile advertiser ids or third party identifiers

AP Cookie: The cookie header in the request, including name-value pairs and the associated attributes.

The cookie headers is used to tie each impression to an AudienceProject panelist with if the impression is initiated from a web-environment. Our panel is owned and operated by Audience Project and build on first party cookies. We don't rely on third party panelists.

If the impression originates from within an native application, the mobile advertiser ID will be included as a parameter identifying the panelist instead of the cookie-id.

With select partners, the panel sync is performed server to server utilising SHA256 mail-hashes and a third party match-table is build instead. In such a scenario, the third party will include the match-table id as part of the event parameter.

It is important to understand that due to the unique 1:1 relation of behavioural data and panelist information in AudienceReport, we are capable of estimating not only reach in real-time, but to measure it across medias and platforms in order to estimate not only reach, but unique reach across different medias and tracking-points. A good example of this capability can be seen below:

As shown in the table - looking across different web platforms, AudienceReport not only see the reach generated by each tracking point / media / line-item, AudienceReport give the unique reach as well, allowing our clients to estimate and optimize against effective CPM's as the campaign progresses.

In order for validation measurement to work, AudienceProjects AudienceReport need:

A: Means of identifying which of our panelists that were exposed to a particular tracking-point.

Options are:

- First party AudienceProject Cookie
- Mobile advertiser ID
- Custom S2S panelist sync initiated with hashed e-mails

B: Event logs containing events with functional columns equal to the columns described above.

If real-time deliveries are not an option, batch intervals should be as short as possible and with as little delay as possible.

Existing TV integrations between AudienceReport and TechEdge requires that data are delivered to AudienceProject with little to no delay in order to ensure full processing and delivery to the TV meter measurement system.

While AudienceProject and AudienceReport have the capabilities of recalculating results in real-time as well as adding delayed data retrospectively, many of our TV integration partners does not have this capability. Once data is delayed beyond a certain point - it will be considered lost due to the batch nature of the TV-meter measurement approach.

Estimation of online reach

In order to be able to estimate online reach we need to estimate several parameters utilizing traditional methods of extrapolation that will be combined with measured full universe data. The following walkthrough have been divided into two parts. a “simple” model that introduces core principles and approaches without taking double-coverage, frequency and incidence-challenges into account. Later we expand from the simple model into a more complex & realistic model that addresses those issues as well.

We will refer to campaigns and line-items through the description since the methodology behind the AudienceReport campaign measurement tool is the same being utilized for performing media-centric measurement.

The simple model explained

Our simple model approach requires us to estimate the following parameters:

Relative reach

The relative proportion of recognized panelists that are in the required target group. The share is estimated by taking the sum of weights on all recognised panelists in target group and dividing with the sum of all weights on all recognized panelists.

$$RelativReach = \frac{\sum_n^{ExpInTargetGroup} Wgt}{\sum_n^{ExpCampaign} Wgt}$$

Unique persons

The number of unique persons exposed to a particular campaign. We know the total number of impressions shown on any given campaign (hard fact). If we combine our knowledge of the number of impressions with an estimate of frequency / person, we will end up with the number of unique persons exposed (opportunity to see) to the campaign in question.

$$Unique\ persons = \frac{Impressions}{Frequency}$$

Using a simple frequency estimation on session-level will usually end up over-estimating the number of unique persons due to not taking cookie-loss into account, the use of multiple devices per person as well as the use of ad-blockers and cookie-deletion protocols like Apple's ITP.

That's why we also estimate the average frequency within our panel, where we can take into account that panelists own and operate multiple devices and where we also have the capability of assessing and estimating the probability of a low individual frequency being the product of not seeing the campaign? Versus being the product of cookie-loss? We will walk through the overall principles elsewhere on how we perform device-deduplication and cookie-loss correction. For now it's just important to understand the principles being the frequency estimation.

Frequency

By taking the sum of the total number of impressions registered on validated panelists only (estimated to have had a sufficient subset of live cookies / device-identifiers throughout the measurement period), we can divide that with the number of unique panelist exposed and get a campaign frequency.

$$Frequency = \frac{\sum_n^{Recognized} Impressions}{Recognized}$$

Total reach

Knowing relative reach and the number of exposed unique persons on any given campaign, we are now able to estimate total reach as:

$$Total\ reach = (RelativReach * Unique\ persons)$$

$$TotalReach = \frac{\sum_n^{ExpInTargetGroup} Wgt}{\sum_n^{ExpCampaign} Wgt} * \left(\frac{Impressions}{\frac{\sum_n^{Recognized} Impressions}{Recognized}} \right)$$

The complex model explained

Trying to estimate online reach creates a series of challenges where the typical TAM approach alone won't suffice. The most obvious issue is the move away from operating with a well-defined and well-known panel-population (TAM) to having to use volatile cookie- and device-identifier based panels in combination with various behavioural data-sources that can't always be tied to panel-members. We will address three different challenges in our walkthrough of the complex model.

Two of the challenges appear when measurement is performed across 2 or more trackingpoint, line items or media that don't exhibit similar characteristics in terms of panelist incidens and/or frequency. The third challenge is tied to estimating double coverage across tracking-points/line items/medias.

Different frequency across different tracking points

Since frequency is part of the foundation of estimating the number of unique person, a difference in frequency across multiple tracking point will therefore result in over/under estimation of the individual tracking-points when joined. Mix low-frequency tracking points (or media) with high-frequency tracking points would result in weighting down the low-frequency tracking points and vice versa if a simple campaign average is utilized. And since different tracking points (or media) typically contribute with different demographic audiences, the use of simple averages would also cause discrepancies in the demographic profiles generation.

In order to facility different frequencies across multiple tracking points, we need to extend the current simple frequency estimation approach:

$$Frequency = \frac{\sum_n^{Recognized} Impressions}{Recognized}$$

Instead we need to estimate the overall frequency as a product of the individual frequency estimates on each tracking point. The collection individual frequencies will afterwards be used to estimate the overall campaign frequency.

$$FrequencyREV = \sum_z^{TrackingPoint} \frac{\sum_n^{Recognized(z)} Impressions(z)}{Recognized(z)}$$

Same approach will be utilized when estimating the unique number of persons exposed. Instead of doing an average on the overall campaign:

$$Unique\ persons = \frac{Impressions}{Frequency}$$

We will estimate the number of unique persons reached on each and every tracking point (media) and sum up the individual metrics to the overall aggregated campaign metric.

$$Unique\ personsREV = \sum_z^{TrackingPoint} \left(\frac{Impressions(z)}{Frequency(z)} \right)$$

Difference incidence rate across different tracking points

A similar challenge is present when we look at incidence rates across tracking points and media. The relative share of recognised panelists is fluctuating across different target groups, device-types and media-types. If we ignore the fluctuations we will essentially accept that high-incidence tracking points (media) will be overrepresented (due to the numerical higher number of recognised panelists) and low-incidence vice-versa.

The solution to this challenge is to estimate the combined campaigns relative reach by first estimating reach individually on each tracking point / media and sum the results into combined campaign reach.

Relative reach revised:

$$RelativeReach = \sum_z \frac{\sum_n^{ExpInTargetGroup(z)} Wgt}{\sum_n^{ExpCampaign(z)} Wgt}$$

Which now allows us to estimate the absolute reach of the campaign by utilizing the revised estimation of relative reach as well as unique persons.

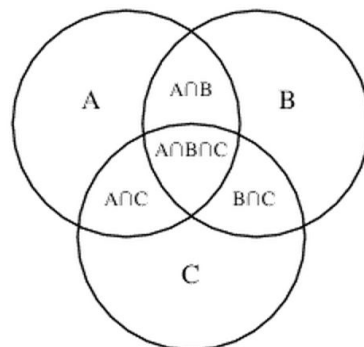
Absolute reach revised:

$$Absolute\ reach = (RelativeReach_{REV} * Unique\ persons_{REV})$$

Double Coverage across tracking points

The third challenge our model needs to account for, is tied to the consequence of cross-exposure across multiple tracking points / media. The model we have been working with so far not does account for the fact that often we will see the same panelist exposed across multiple tracking-points across the very same campaign. Utilizing standard models would in the case of double exposure result in the double exposed panelists to carry additional weight within the model.

A campaign with a few tracking points / media are relatively manageable. Three tracking points only gives us 7 different alternatives for double-exposure for any single panelist.



But the challenge grows each campaign is capable of measuring 256 different tracking points / media placement and/or subsections. Resulting in each and every exposed panelist can be exposed in a subset of the 256 tracking points and thereby in theory be part of

$$N = \sum_{k=0}^n \binom{n}{k} = 2^n$$

Different exposure profiles (n=number of tracking points). Which in principal generates up to 2^256 possible combinations of double coverage profile per panelist per campaign.

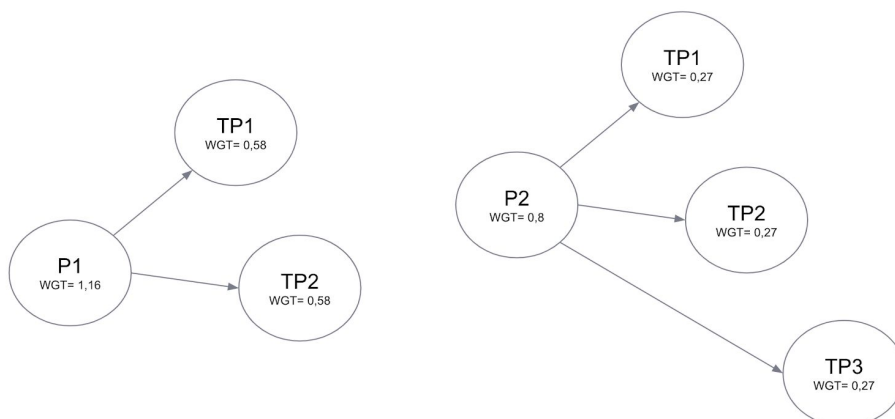
As it is not uncommon to recognise between 10.000-30.000 panelist on national campaigns and every panelist end up with any one of 2^256 double coverage profiles, trying to create a correction matrix would be a very serious challenge.

The solution we chose instead was to decompose the individual panelists weight (WGT) across the tracks points where the panelists have been exposed. The overall weight is being split into “shards”. The sharded weight is used afterwards as the foundation for estimating relative reach on trackpoint level. The result is the individual panelist only counts as 1 X WGT on the overall campaign level no matter how many different tracking point the panelists have been exposed / sharded across.

The principle is illustrated in the table and drawing below:

<i>UID</i>	<i>Exposure</i>	<i>Wgt(panelist)</i>	<i>Shard(Wgt)</i>
387-829	$A \cap B$	1,77	$1,77/2 = 0,885$
901-661	$B \cap D$	0,92	$0,92/2 = 0,46$
571-992	$A \cap B \cap D$	1,12	$1,12/3 = 0,373$
982-223	B	2,074	2,074

Another way to illustrate it from a Panelist perspective vs. exposed tracking points / media



The relative reach estimation will utilize the specific shards generated for the specific campaign based on each panelist exposure across different tracking points / media.

$$\text{RelativeReach} = \sum_z \frac{\sum_n^{\text{ExpInTargetGroup}(z)} \text{Shard}(Wgt)}{\sum_n^{\text{ExpCampaign}(z)} \text{Shard}(Wgt)}$$

Cross-device management

When we refer to frequency and a panelist, we are referring to the construction of a Panelist-ID. Each panelist contains a collection of device-id's that can be a collection of both physical as well as virtual id-universes.

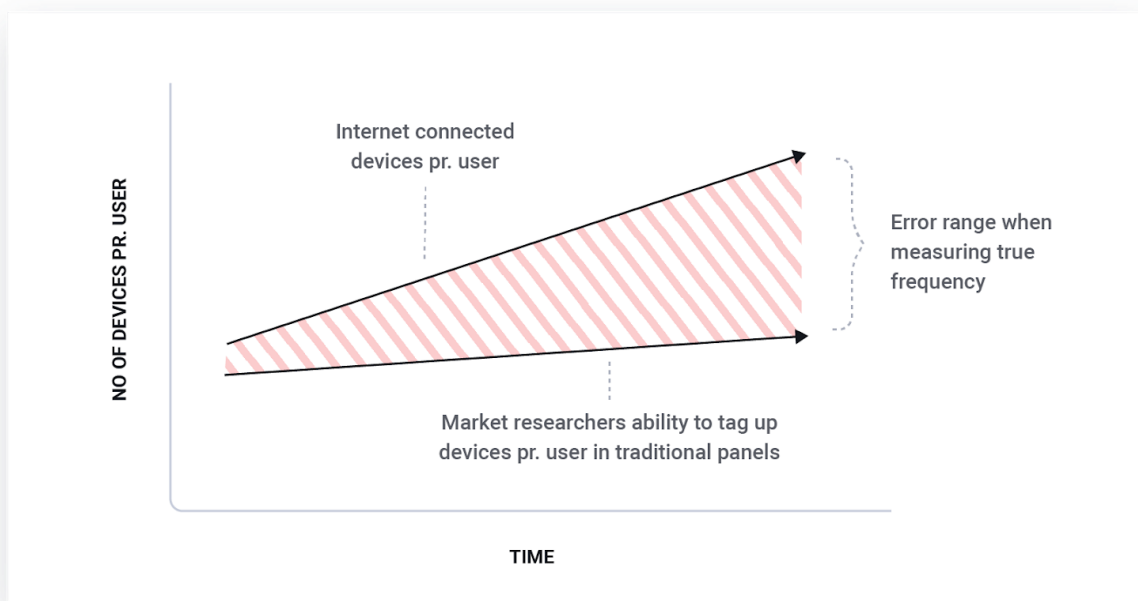
Tagging, measuring and aggregating device id's into a collection of human-id (or panelist id's) is a significant challenge that requires its own explanation. We will discuss and explain some of these challenges in more detail below in "Getting Frequency Right".

Getting Frequency right

Key challenges to frequency measurement

The futility of thinking only in market research terms is perhaps best understood when dealing with the challenge of estimating true frequency, which really goes to the heart of a core issue the market research industry faces when trying to keep track of the development in online usage. Therefore, we will elaborate on this below to help facilitate a deeper understanding of our system.

The key issue is that more and more people are using more and more devices and the growth in the ability to tag devices using traditional Market Research means cannot keep up:



The problem is a hard one. Below is a more thorough explanation of some of the problems using traditional cookie based methods - which still pertains even if more and more companies claim to have solved the “cookie loss problem”¹.

Measuring online frequency correctly is a tricky business. The problem is not new: regular web analytic tools usually fail to calculate a correct frequency due to a lack of methodical understanding of the challenge that lies ahead and a lack of technical ability to de-duplicate from devices to humans.

It is therefore not unheard of to see websites boast that they manage to attract more visitors than the actual population of the country or even the world. The Washington Redskins managed to boast

¹ We here take for granted that the concept of frequency measurement by recruiting a small panel of respondents that are required to install software trackers on all of their devices is not sustainable. It must be some form of passive measurement that becomes the solution in order to accurately measure long tail traffic.

attracting an audience of 7 billion unique visitors back in 2015². Which is either pretty impressive or wrong on so many levels.

In order to get online frequency (or unique visitors) right, we need to understand the basics of the frequency estimates.

In traditional web analytics each impression or pageview will be registered as a unique log-line for further analysis. But in order to determine whether 1.000.000 log-lines were generated by 1.000.000 unique devices or rather 100.000 unique devices each generating 10 impressions, we need to tie together the individual log-lines.

The de-facto standard for tying together log-lines are cookies. If a cookie has been stored on the visitor's computer, the individual log-lines will show the user-id of the particular device generating the impressions. We can then group the impressions on user-id and estimate an average frequency. Cookies have however massive challenges, which all result in the number of unique visitors being overestimated.

Challenge 1: Cookie-death

The first challenge is the fact that cookies are not omnipotent identifiers that last forever. Cookies are being deleted all the time. Either by users or due to expiration.

When a cookie is deleted we lose the ability to tie past impressions to future impressions. And worst of all, it is not possible to detect 'when' a cookie has been lost. Deleting a cookie is a passive operation that happens within the users' browsers. The website that originally set the cookie will not be notified about the deletion.

So whenever you stop registering a particular cookie-id on your website or campaign, it could be either due to the user not visiting any more or due to the cookie being deleted. There is no (simple) way to tell if it was abandonment or death.

The impact of cookie-death on any attempt to estimate a correct frequency is profound. Not only will cookie deletion result in the inability to tie impressions together for that particular device, but it will also result in a new cookie ID being issued to the same device, thus resulting in even more misleading metrics.

Assume the same device accessing a website daily for 21 days:

Unique visitors = 1
Average Frequency = 21

Now, let's presume the user's cookie was deleted after 7 days and again after 14 days. Every time the user arrives at the website in question with no cookie (the old being deleted) a new one is issued.

Instead of registering 1 unique device with a frequency of 21, we now have 3 different cookie-id's each with 7 impressions registered. In other words:

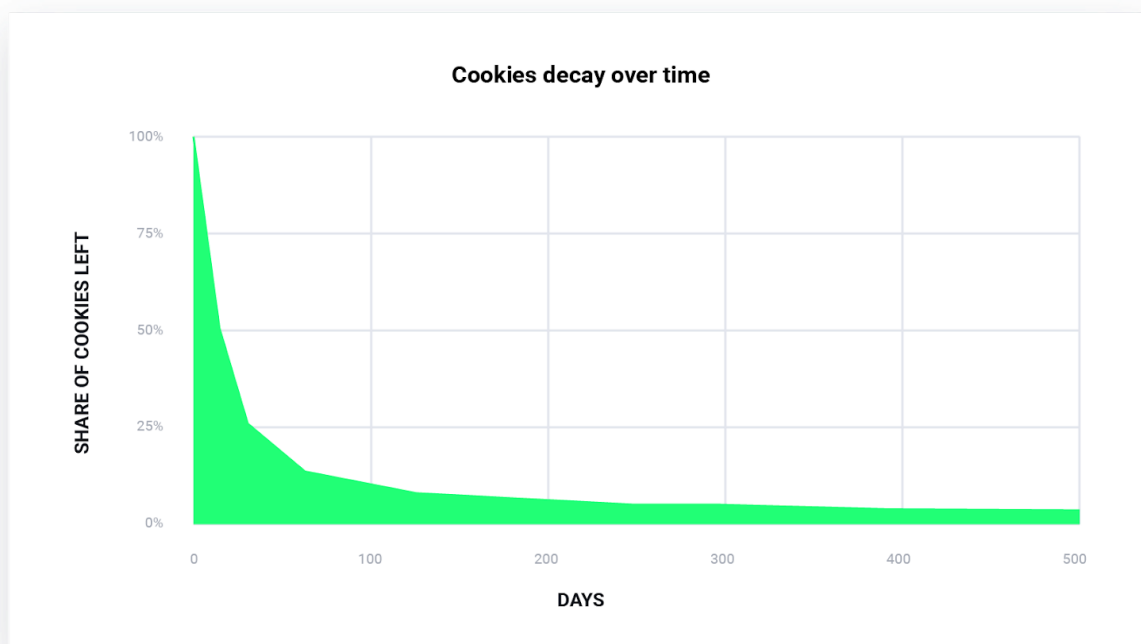
2

<https://www.washingtonpost.com/news/dc-sports-bog/wp/2015/07/29/no-every-person-on-earth-did-not-read-about-the-2014-re-dskins-training-camp/>

Unique visitors = 3
Average Frequency = 7

Our estimated frequency now displays an error that is a factor of 3 on both counts. Both unique visitors as well as frequency. The measurement error will increase dramatically over time due to cookie-loss being a function of time. The longer we try to measure a website or a campaign, the bigger the risk that more cookies are lost.

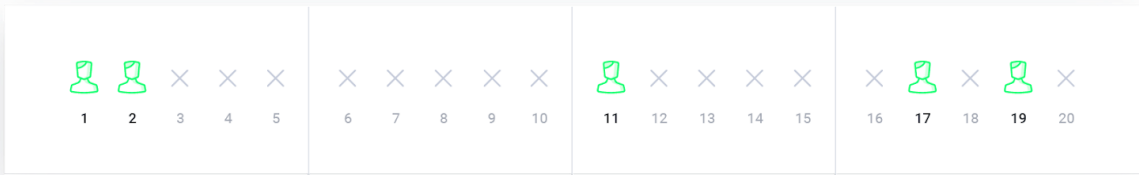
Cookie-loss over time can be defined by using the concept of half-time. Normal cookie-based panels have a half-time of 16 days. Meaning that within 16 days, on average 50% of the cookie-population have been lost.



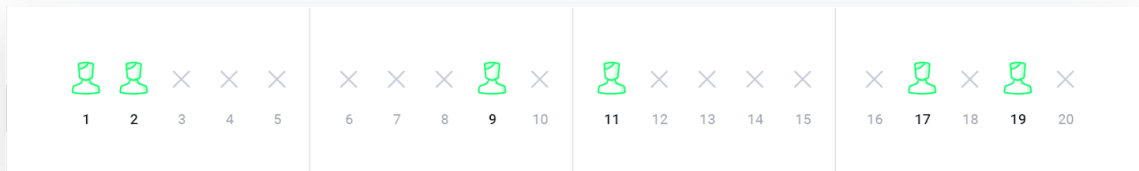
The impact of the decay is two-fold. When a cookie is lost we will stop accumulating impressions on that particular user/device. Which result in an underestimation of frequency. At the same time cookie-less users will usually be re-tagged under a new cookie-id. Further impacting both frequency and the unique user count.

Given a cookie half-time of 16 days we can estimate how many unique visitors the Redskins actually had if traffic was tracked over, say, a year. Let c be the cookie half-time in days, let d be the number of days we track traffic, and let U be the true number of unique visitors. U is the number we want to estimate.

We assume that all unique visitors are seen at least once in the first cookie half-time period, i.e., the first c days, and then each unique visitor is seen at least once in each subsequent period. For example, if $c = 5$ and $d = 20$, the pattern of a visitor could look like this:



but not:



Since the visitor is not seen in the second period.

The reported number of unique visitors \hat{U} is then:

$$\hat{U} = U + \frac{1}{2} U \left(\frac{d}{c} - 1 \right) \quad (1)$$

Basically, this accounts for losing half of the cookies in every period. This half is then re-introduced as new unique visitor.

Solving (1) for U we get:

$$U = \frac{2c\hat{U}}{d+c}$$

Assuming the Redskins story was measured over a period of a year. Setting $d = 365$ and $c = 16$ we get that the actual number of unique visitors is:

$$U = 658.936.307$$

This is far more reasonable compared to the 7.845.460.401 reported. However, due to their very aggressive way of counting exposure, the real number is probably a lot smaller.

Challenge 2: Ad-blockers / cookie-rejects

Another issue that needs to be addressed is the fact that an increasing share of online users are deploying various ad-blockers that wipe cookies or straight out refuse to accept cookies at all.

The result is that a certain percentage of users will lose their cookies as soon as their browsing session ends. In such scenarios the cookie-loss will be 100% within a 24 hour timeframe. The result is that a single user with such an Adblock/browser configuration can be counted multiple times.

A single user that visits a website on a daily basis and generates 4 pageviews per/day on average will under normal circumstances be counted as 1 unique, Frequency = 4 x 31 days = 124.

But with the 24 hour deletion the user will all of a sudden be counted as 31 unique users with an average frequency of 4. Resulting in the unique count being overestimated by a factor of 31. And the frequency being underestimated by a factor of 31 as well.

The effect of Adblockers/Browser-cookies blocking is not uniform across different websites. AudienceProject has extensively documented the usage of Adblockers and similar technologies amongst different parts of the online population³.

The problem with ad blockers in combination with uneven cookie loss rates across different demographics segments makes it very difficult to deduplicate using some kind of a fixed ratio between the number of observed cookies and 'true frequency' (which is a fairly standard deduplication method by many market research providers). Also, it makes the whole concept of panel size somewhat dubious. What do we mean by panel size? It is as necessary to have a stable, ongoing, inflow of new panelists as it is to have a large panel-base at any given point in time. A panel is more of a flow-concept than a stable entity.

Challenge 3: First party versus third party

Another problem specifically related to many ad-servers and online analytics services is that they often try to measure frequency and users utilizing their own id-universe. Ad-servers often rely on their own cookie management system in order to execute campaigns and to measure their online effect. When a third-party cookie system is used for setting and reading cookie-id's special precautions need to be taken in order to ensure correct measurement.

iOS devices like iPhones and Ipads do not allow third parties to set cookies by default. The same goes for newer versions of Safari running on MacOS, as well as several versions of the Firefox browser. And with the introduction of the latest ITP iterations, even first party cookies can be classified and demoted to third party cookies.

If no corrective action is taken, a large part of especially the mobile device population and laptops used predominantly amongst the younger cohorts (Macs) will not contribute to the visitor estimates or even skew the estimates in the wrong direction.

In addition, as increasingly more devices and applications (such as in-app, gaming consoles, Apple TV etc.) which do not support any form of cookies are used for video and other activities where measurement is needed, other measures are desperately needed.

3

<https://www.audienceproject.com/blog/key-insights/new-study-uk-us-ad-blockers-want-relevant-ads/>

<https://www.audienceproject.com/blog/key-insights/new-study-ad-blocking-decreases-in-denmark/>

What can be done?

AudienceProject is in a unique position to estimate and validate day-to-day cookie-loss by deploying a combination of daily large scale data analysis of billions of loglines of data and deterministic data from our extensive online research panels. Has one of our cookies been deleted or is it indeed a new user? To figure this out, we rely on a combined approach.

Research panels

AudienceProject owns and operates the biggest online email based research panels in the UK, Germany, the Nordics and elsewhere. Every month we complete more than 150.000 interviews across our panels. Since invitations within these research panels are issued by email hashes, we don't have to rely on cookies for identification. But using the email panels in combination with cookie id's we can determine cookie-loss rates within different segments of the online population and monitor them in real time. This approach is only possible to deploy if you have access to vast operational panels that utilize persistent identifiers that are not cookie-based - along with emails, we also support IDFAs, advertiser-id and basically all other persistent identifiers that enter our eco-system. They enrich our ability to provide extremely powerful identity management (and therefore cross-device deduplication).

AudienceProjects big data approach

Another source of information is AudienceProject's extensive access to online behaviour. Every day AudienceProject receives and analyses several billion events received from websites and campaigns across the countries in which we operate.

The data provides AudienceProject with unique insights into solving the cookie-retention issues. Remember that when a cookie-ID stops appearing on a website or campaign, we can't really know if it is due to the cookie being deleted or the user just stopping to visit the website?

AudienceProject performs an extensive 90 day analysis across hundreds of billion impressions and events registered across different websites and online properties (in the form of loglines of data). Using the signals in the data, we can estimate the probability that any given cookie-ID has been lost with a very high degree of accuracy, due to our ability to train our algorithms against our deterministic dataset (this dataset being our core panel where data are deterministic).

Using these vast amounts of data and a bayesian approach, AudienceProject can ensure that cookie loss does not contaminate the frequency estimates and thus the unique exposure counts that are the foundation for estimating reach.

This enables us to deal with another problem. Cookie-loss is not a uniform effect across all users and websites. Different demographic target groups exhibit very different decay patterns. A classic example is young males' tendency to delete cookies extremely often. If we fail to account for the fact that different audiences carry different attributes and that different websites, media and campaigns attract different audiences, we risk carrying the inherent measurement errors into our audience analytics services.

The key question is: How do we go from device frequency to human frequency?

Humans. Not devices.

Device-deduplication is complex. To take an example, we have decomposed our Danish audience panel into some submetrics:

“Looking at the past 6 months, AudienceProject measured more than 50 million (52.064.557 at last count) different device id’s in Denmark alone. Some device ids are seen only a few times but the vast majority (> 75%) are seen more than 10 times. 7.651.479 of the device ids can be resolved as devices that belong to 3.210.884 distinct humans living in one of 1.762.454 identified households. In other words, we are measuring ~85% of the total online population in Denmark and we see the same person on an average of 2.38 device ids.”

A single smartphone will often appear under several different id’s. First and foremost the native browser will have its own identity, whereas web pages entered through Facebook, Pinterest, Quora or another social media app will each appear with their own identity due to the fact that such web pages are not loaded within the smartphone’s native browser, but through a browser-plugin build into the app itself.

On top of that, we have a whole collection of what appears to be native ads but in reality are not which create problems when trying to tie these ads to the native browser identity. Device deduplication should therefore, more properly, be called application deduplication in that the same device can have more id’s than one.

How does the ad tech industry go about cross-device mapping?

Within the industry, cross-device estimation is often based on one of two competing approaches. Major vendors with direct access to a large proportion of the online population through first party interaction, either through user-logins or e-mails, will often go for a deterministic approach. Many third-party vendors that lack the direct user-interaction will opt for the probabilistic approach.

The two different approaches each come with their trade-off, primarily between accuracy and scale.

The deterministic approach

With this approach first-party identifiers like login-information, e-mail addresses, e-commerce transactions or social media transactions are used in order to determine whether a particular person is in fact using multiple devices.

The deterministic approach is based on first-party data since data related to logins, emails etc. needs to be collected from the company that provides the service that acts as the identifier. If someone logs in on the same account from multiple devices, the company providing the account will be the entity providing the first party deterministic data about the different devices belonging to the same person.

While deterministic first party data provides a very high degree of accuracy, it is by no means perfect. Shared devices still pose a challenge even when using the deterministic data. The person operating the device might not always be the person logged in. But overall the levels of accuracy are reasonably high. The deterministic approach’s main challenge is that it suffers from only being able to offer a very limited view of a vast and complicated universe. It is in effect a small non-random sample of a very big universe. We might obtain strong indicators about relations within the deterministic data, but are still left in the dark when it comes to identifying unknown unknowns. Therefore, such data are not particularly well suited for calibrated audience measurements aiming to deliver a representative view of the universe.

When a user only appears to own one device according to deterministic login data, how can we tell whether it is the truth or whether we fail to capture the multiple devices with our limited pool of deterministic data?

Scale is the true Achilles heel of the deterministic approach. There are very few companies in the world (aside from Facebook) that can document a sufficient scale in their active user base in order to provide a comprehensive overview of the whole cross-device universe.

The probabilistic approach

The probabilistic data approach is typically based on behavioural data that are aggregated and analysed in order to determine the probability of two or more devices being related.

Advanced algorithms try to identify behavioural patterns like similar travel and browsing behaviour in order to determine if several devices belong to the same person. Many probabilistic models are in fact searching for distinct patterns of known human behaviour, patterns emerging due to humans being creatures of habit.

- **The majority of people** go to work or school and tend to carry their phone/laptop
- **The majority of people** have a concept of “normal business hours”
- **We share our internet connection** with family members and/or colleagues
- **We do not go to work on weekends**, almost never on official holidays
- **Our spare time** is normally spent either at home or visiting other homes.

All these habits create distinct behavioural patterns that often can be identified algorithmically by analyzing anonymised log-files.

The advantage of using probabilistic modelling is the ability to scale your models, since you no longer have to rely on first party interactions and people providing you with login information like usernames and email addresses across all of their devices. The probabilistic approach also allows a wider range of devices to be included. If a user browses from the built-in web browser in a Tesla model S, the behaviour can in theory be observed, logged and matched to a specific household, smartphone or laptop.

The true strength of the probabilistic approach lies in its ability to scale, but its inherent weakness is a lack of deterministic data to actually validate the accuracy of the model.

AudienceProject’s approach to cross device

AudienceProject deploys a combined approach where probabilistic modelling is used to map relations between all devices, known as well as unknown, while the deterministic data is used for testing accuracy and improving our behavioristic models iteratively.

Going back to the model above, where we showed the increasing discrepancy between the ability to tag devices per user and the growth of devices per user, we have solved this dilemma by using our deterministic dataset as a base on which to train models that use our vast amounts of behavioral data to associate unknown devices to our panelists.

This approach gives us the benefit of high accuracy levels combined with massive scale.

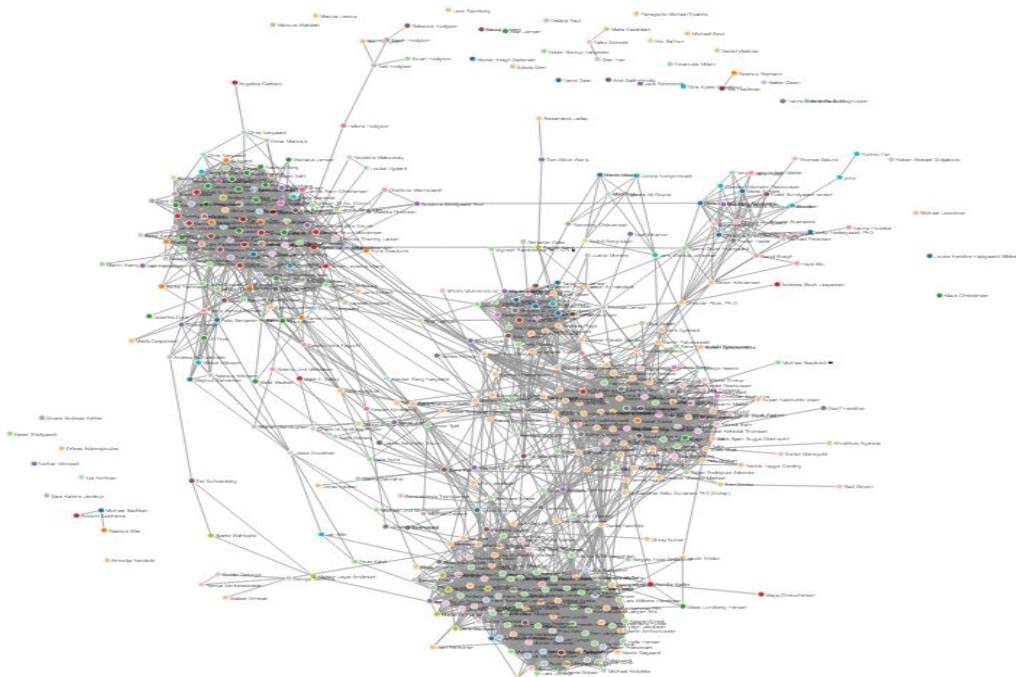
The probabilistic model deployed in order to identify cross device interaction is based on our proprietary knowledge graph, the AudienceGraph, which is built on graph analytics, a different approach to relation analysis than normally deployed on more regular types of data. But what are the inherent advantages of building a device-graph instead of utilizing regular relational methods?

- **Relational analytics** typically explore relationships by comparing them “one-to-one” or “one-to-many”
- **Graph analytics** is different from traditional relational analytics by exploring “many-to-many” relationships

If we use social networks like Facebook and LinkedIn as an example (who also use knowledge graph technologies similar to what is applied by AudienceProject), it would be quite easy to identify a single account and the accounts of 10 friends and/or next level connections using traditional relational analytics. It would also be quite easy to find any number of accounts and all of their friends/connections in a few orders. But it will be next to impossible to move beyond this level of complexity in associations.

Graph analytics on the other hand is capable of comparing “many-to-many” relations. With relational analytics it is very difficult to answer any questions about the second level of “indirect” connections or friends an account has. Graph analytics on the other hand make it possible to answer questions not only about the connections and friends of a person but also about all of their friends and connections.

This approach allows us to identify clusters within a social network that could constitute colleagues or family, solely by analysing the level of indirect connections within each of these clusters of connections. A type of connection that is difficult to identify relying on relational analytics only.



To sum up:

- **Relational analytics** are well suited for analysing structured deterministic data that can be represented in classical tables consisting of rows and columns. Good for analyzing traditional panel/user-data for small panels with a limited, stable, number of devices (one TV screen, for example)
- **Graph analytics** are well suited for analysing unstructured data that is in a constant state of flux. Good for analysing social networks and online behaviour - and for tying devices to our panelists to fill the gap between a panelists true number of devices and the number, we know deterministically.

AudienceProject uses graph analytics in order to determine the number of devices that are related and belong to a specific panelist member.

The ability of the graph to tie devices to panelists is nothing but astoundingly precise. When training the model against deterministic data (where we know the relationship between a panelist and device for sure), the accuracy levels average 95-96 %. Is a clear example of why it is simply necessary to marry sophisticated technology with panels to achieve proper measurement in this digital age.

Appendix A - Detailed description of sources for universes

Use of information and communications technology by individuals

Producer: Statistics Finland

Homepage: [http:// www.stat.fi/til/sutivi/index_en.html](http://www.stat.fi/til/sutivi/index_en.html)

Main topic: Science, Technology and Information Society

Related topics: Culture and the Media

Official Statistics of Finland (OSF): Yes

European Statistical System (ESS): Yes

Description

The purpose of the survey on Use of information and communications technology is to produce data about ICT usage in households and by individuals. The data are used for the development projects of Finnish information society and for compiling pan-European information society indicators.

Data content

The objective is to examine the generality of people's computers and Internet use, use purposes and consumers' net commerce. The survey results are available, the basic data are confidential.

Classifications used

The classification used include household size and structure, age, gender, regional classifications based on the division of municipalities.

Data collection methods and data sources

The survey is a sample survey carried out annually in spring and summer as a telephone interview and online among the population aged 16 to 89.

Data collections

Use of information and communications technology

Updating frequency

Once a year

Time of completion or release

Third or fourth quartile

Time series

Varies by time series, most time series at least from the year 2002 onwards.

Keywords

communication technology, computers, data protection, data security, domestic appliances, electronic commerce, information society, information technology, internet, mobile phones, online studying, social media, telephones

Source tables:

<https://www.stat.fi/til/sutivi/tau.html>

NB: Access in Finnish to see full overview of available tables. English version is limited in content.

Appendix B - Content of columns in W3C extended log files

Field Number	Field Name	Description
1	date	The date on which the event occurred in the format yyyy-mm-dd, for example, 2015-06-30. The date and time are in Coordinated Universal Time (UTC).
2	time	The time when the CloudFront server finished responding to the request (in UTC), for example, 01:42:39.
3	edge-location	The edge location that served the request. Each edge location is identified by a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location. (These abbreviations might change in the future.) For a list of edge locations, see the Amazon CloudFront detail page, http://aws.amazon.com/cloudfront .
4	sc-bytes	The total number of bytes that CloudFront served to the viewer in response to the request, including headers, for example, 1045619.
5	c-ip	The IP address of the viewer that made the request, for example, 192.0.2.183 or 2001:0db8:85a3:0000:0000:8a2e:0370:7334. If the viewer used an HTTP proxy or a load balancer to send the request, the value of c-ip is the IP address of the proxy or load balancer. See also X-Forwarded-For in field 20.
6	cs-method	The HTTP access method: DELETE, GET, HEAD, OPTIONS, PATCH, POST, or PUT.
7	cs(Host)	The domain name of the CloudFront distribution, for example, d111111abcdef8.cloudfront.net.
8	cs-uri-stem	The portion of the URI that identifies the path and object, for example, /images/daily-ad.jpg.

9	sc-status	<p>One of the following values:</p> <ul style="list-style-type: none"> • An HTTP status code (for example, 200). For a list of HTTP status codes, see RFC 2616, Hypertext Transfer Protocol—HTTP 1.1, section 10, Status Code Definitions. For more information, see How CloudFront Processes and Caches HTTP 4xx and 5xx Status Codes from Your Origin. • 000, which indicates that the viewer closed the connection (for example, closed the browser tab) before CloudFront could respond to a request. • If the viewer closes the connection after CloudFront starts to send the object, the log contains the applicable HTTP status code.
10	cs(Referer)	<p>The name of the domain that originated the request. Common referrers include search engines, other websites that link directly to your objects, and your own website.</p>
11	cs(User-Agent)	<p>The value of the User-Agent header in the request. The User-Agent header identifies the source of the request, such as the type of device and browser that submitted the request and, if the request came from a search engine, which search engine. For more information, see User-Agent Header.</p>
12	cs-uri-query	<p>The query string portion of the URI, if any. When a URI doesn't contain a query string, the value of cs-uri-query is a hyphen (-).</p> <p>For more information, see Configuring CloudFront to Cache Based on Query String Parameters.</p>
13	cs(Cookie)	<p>The cookie header in the request, including name-value pairs and the associated attributes. If you enable cookie logging, CloudFront logs the cookies in all requests regardless of which cookies you choose to forward to the origin: none, all, or a whitelist of cookie names. When a request doesn't include a cookie header, the value of cs(Cookie) is a hyphen (-).</p> <p>For more information about cookies, see Configuring CloudFront to Cache Objects Based on Cookies.</p>

14

edge-result-type

CloudFront classifies the response after the last byte left the edge location. In some cases, the result type can change between the time that CloudFront is ready to send the response and the time that CloudFront has finished sending the response. For example, in HTTP streaming, suppose CloudFront finds a segment in the edge cache. The value of `x-edge-response-result-type`, the result type immediately before CloudFront begins to respond to the request, is Hit. However, if the user closes the viewer before CloudFront has delivered the entire segment, the final result type—the value of `x-edge-result-type`—changes to Error.

Possible values include:

- Hit – CloudFront served the object to the viewer from the edge cache.
- For information about a situation in which CloudFront classifies the result type as Hit even though the response from the origin contains a Cache-Control: no-cache header, see [Simultaneous Requests for the Same Object \(Traffic Spikes\)](#).
- RefreshHit – CloudFront found the object in the edge cache but it had expired, so CloudFront contacted the origin to determine whether the cache has the latest version of the object and, if not, to get the latest version.
- Miss – The request could not be satisfied by an object in the edge cache, so CloudFront forwarded the request to the origin server and returned the result to the viewer.
- LimitExceeded – The request was denied because a CloudFront limit was exceeded.
 - CapacityExceeded – CloudFront returned an HTTP 503 status code (Service Unavailable) because the CloudFront edge server was temporarily unable to respond to requests.
- Error – Typically, this means the request resulted in a client error (sc-status is 4xx) or a server error (sc-status is 5xx).
 - Redirect – CloudFront redirects from HTTP to HTTPS.
- If sc-status is 403 and you configured CloudFront to restrict the geographic distribution of your content, the request might have come from a restricted location. For more information about geo restriction, see [Restricting the Geographic Distribution of Your Content](#).
- If the value of `x-edge-result-type` is Error and the value of `x-edge-response-result-type` is not Error, the client disconnected before finishing the download.

15	edge-request-id	An encrypted string that uniquely identifies a request.
16	x-host-header	<p>value that the viewer included in the Host header for this request. This is the domain name in the request:</p> <ul style="list-style-type: none"> • If you're using the CloudFront domain name in your object URLs, such as <code>http://d1111111abcdef8.cloudfront.net/logo.png</code>, the x-host-header field contains that domain name. • If you're using alternate domain names in your object URLs, such as <code>http://example.com/logo.png</code>, the x-host-header field contains the alternate domain name, such as <code>example.com</code>. To use alternate domain names, you must add them to your distribution. For more information, see Using Alternate Domain Names (CNAMEs). • If you're using alternate domain names, see <code>cs(Host)</code> in field 7 for the domain name that is associated with your distribution.
17	cs-protocol	The protocol that the viewer specified in the request, either <code>http</code> or <code>https</code> .
18	cs-bytes	The number of bytes of data that the viewer included in the request (client to server bytes), including headers.
19	time-taken	number of seconds (to the thousandth of a second, for example, 0.002) between the time that CloudFront edge server receives a viewer's request and the time that CloudFront writes the last byte of the response to the edge server's output queue as measured on the server. From the perspective of the viewer, the total time to get the full object will be longer than this value due to network latency and TCP buffering.

20	x-forwarded-for	<p>If the viewer used an HTTP proxy or a load balancer to send the request, the value of x-forwarded-for in field 17 is the IP address of the proxy or load balancer. In that case, c-ip is the IP address of the viewer that originated the request.</p> <p>If the viewer did not use an HTTP proxy or a load balancer, the value of x-forwarded-for is a hyphen (-).</p> <p style="text-align: center;">Note</p> <p>The X-Forwarded-For header contains IPv4 addresses (such as 192.0.2.44) and IPv6 addresses (such as 2001:0db8:85a3:0000:0000:8a2e:0370:7334), as applicable.</p>
21	ssl-protocol	<p>If cs-protocol in field 17 is https, the ssl-protocol that the client and CloudFront negotiated for transmitting the request and response. When cs-protocol is http, the value for ssl-protocol is a hyphen (-).</p> <p style="text-align: center;">Possible values include the following:</p> <ul style="list-style-type: none"> • SSLv3 • TLSv1 • TLSv1.1 • TLSv1.2

22	ssl-cipher	<p>When cs-protocol in field 17 is https, the SSL cipher that the client and CloudFront negotiated for encrypting the request and response. When cs-protocol is http, the value for ssl-cipher is a hyphen (-).</p> <p>Possible values include the following:</p> <ul style="list-style-type: none">• ECDHE-RSA-AES128-GCM-SHA256• ECDHE-RSA-AES128-SHA256• ECDHE-RSA-AES128-SHA• ECDHE-RSA-AES256-GCM-SHA384• ECDHE-RSA-AES256-SHA384• ECDHE-RSA-AES256-SHA• AES128-GCM-SHA256• AES256-GCM-SHA384• AES128-SHA256• AES256-SHA• AES128-SHA• DES-CBC3-SHA• RC4-MD5
----	------------	--

23	x-edge-response-result-type	<p>CloudFront classified the response just before returning the response to the viewer. See also <code>x-edge-result-type</code> in field 14.</p> <p>Possible values include:</p> <ul style="list-style-type: none"> • Hit – CloudFront served the object to the viewer from the edge cache. • RefreshHit – CloudFront found the object in the edge cache but it had expired, so CloudFront contacted the origin to verify that the cache has the latest version of the object. • Miss – The request could not be satisfied by an object in the edge cache, so CloudFront forwarded the request to the origin server and returned the result to the viewer. • LimitExceeded – The request was denied because a CloudFront limit was exceeded. • CapacityExceeded – CloudFront returned a 503 error because the edge location didn't have enough capacity at the time of the request to serve the object. • Error – Typically, this means the request resulted in a client error (sc-status is 4xx) or a server error (sc-status is 5xx). <ul style="list-style-type: none"> • Redirect – CloudFront redirects from HTTP to HTTPS. • If sc-status is 403 and you configured CloudFront to restrict the geographic distribution of your content, the request might have come from a restricted location. For more information about geo restriction, see Restricting the Geographic Distribution of Your Content. • If the value of <code>x-edge-result-type</code> is Error and the value of <code>x-edge-response-result-type</code> is not Error, the client disconnected before finishing the download.
24	x-http-version	<p>The HTTP version that the viewer specified in the request. Possible values include HTTP/0.9, HTTP/1.0, HTTP/1.1, and HTTP/2.0.</p>

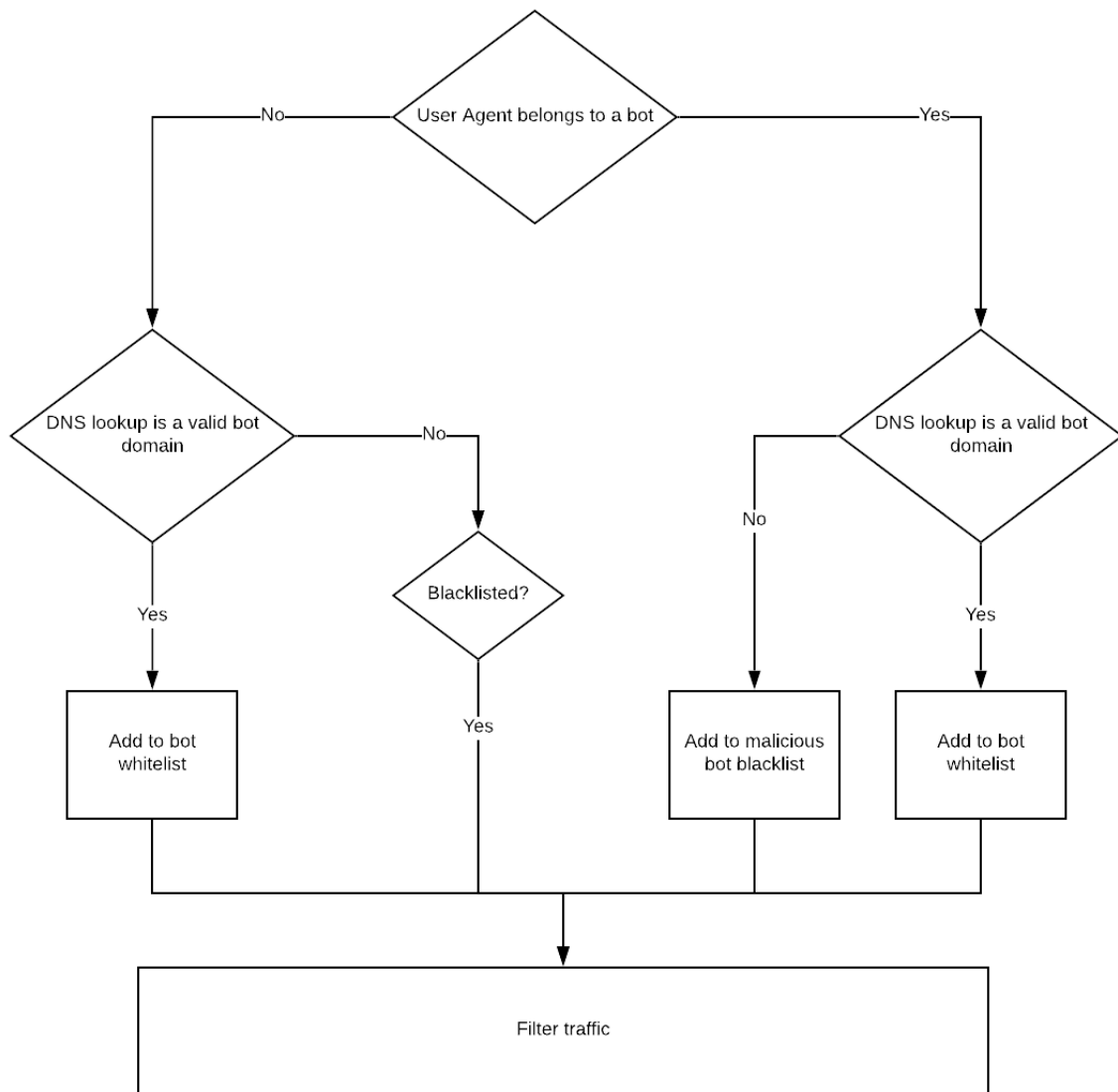
Appendix C - Identification of non-human traffic

Although not all bots are designed to inflate traffic with the purpose of siphoning money from the ad industry, both friendly and malicious bots have as an effect the generation of bogus impressions on visited websites.

The AudienceReport non-human traffic detection service can identify both classes of bots.

- a) Friendly bots - also known as crawlers or spiders. They are scripts which recursively traverse webpages with the purpose of indexing the Internet for search engines. These software entities are usually transparent and they identify themselves through the User-Agent string in the request they make to the server. For example, the following name belongs to the desktop version of Googlebot: *Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)*
- b) Malicious bots: are scripts that can automatically do one of the following:
 - i) Scrape websites to copy content and reuse it in other locations, or to collect data about individuals or businesses
 - ii) Click on banners or download content to generate fake engagement data. These bots purposefully skew analytics data and can affect the decisions of marketers who rely on pay-per-click campaigns.
 - iii) Spam comment threads or email addresses with auto-generated content.
 - iv) Emulate the behavior of legitimate users, in order to crack through online security measures. They could also coordinate to bring down a website through denial of service attacks.

Malicious bots can masquerade as friendly bots by changing their User Agent signature. To detect such misbehaving bots, AudienceReport further validates their identity by verifying that the IP address belongs to official search engine IP ranges. When such IP ranges are not available, such as in the case of Googlebot, a method of connecting the IP address with its domain is used (DNS lookup). Below is a logical flow of the process:



Because DNS lookups are expensive operations, but bot IP ranges fluctuate in time, AudienceReport caches the already identified bot IPs for a limited period of time into a separate friendly bot whitelist and a malicious bot blacklist.

Both the blacklist and the whitelist are populated based on the logic above and sit as the foundation for filtering out traffic in site-centric measurement.

However, the blacklist is further populated with smarter, more ubiquitous bots through the daily analysis of log data in the AudienceReport ecosystem. This is relevant because malicious scripts have a distinct surfing pattern, compared to humans:

- They do parallel HTTP requests at a high rate
- They visit a large number of URLs and tend to cover more routes on a website in a short period of time, compared to humans
- They prefer specific HTTP methods and file type

Appendix D - Overview of identifier lifecycle

Cookie lifecycle

Creation

AudienceProject cookies are created through our free online survey offering that companies, organizations and other website holders can set up to evaluate their users' satisfaction with their website.

At the end of the survey, the respondent is asked if they want to become a member of AudienceProject online survey panel.

You can take a test survey at this link: https://www.userreport.com/#_urp=test_invite

Please note that invitation overlay ensures that AudienceProject has first-party interaction with all our panelists.

This recruitment method has several advantages in creating a high level of representativeness in the panel and consequently in the surveys carried out in the panel. For more details on how AudienceProject ensures this, please revisit the [Panel recruitment](#) section.

Activation and contribution to measurement

In order for a cookie to qualify as an active panelist, two conditions have to be met:

- The cookie should have associated age and gender information, because these two pieces of information sit at the base of the [panel weighing](#) process.
- The AudienceProject system has to have seen the panelist within the last 90 days on one or more of the ad campaigns or websites that we measure.

Once a cookie has been incorporated in the panel and assigned a weight, it will be used as part of the measurement methodology every time it is seen on a tracked AudienceReport campaign. As mentioned in the measurement overview section, AudienceReport measures campaign performance by the application of a single 1x1 SSL tracking pixel (<https://visitanalytics.userreport.com/hit.gif>) to each campaign or placement within a website. The pixel is hosted on AWS Cloudfront ensuring worldwide scalability and low latency.

When the pixel is fired, AudienceReport logs the request in W3C extended log file format, which contains the following columns (see appendix B for a description of the content of each columns of the log files):

date, time, x-edge-location, sc-bytes, c-ip, cs-method, cs (Host), cs-uri-stem, cs-status, cs (Referer), cs (User-Agent), cs-uri-query, cs (cookie), x-edge-result-type, x-edge-request-id, x-host-header, cs-protocol, cs-bytes, time-taken, x-forwarded-for, ssl-protocol, ssl-cipher, x-edge-reponse-result-type and cs-protocol-version

As you can see, the cookie information (if previously set in the first party context of our survey tool) will be available in the request under the `cs(cookie)` field. The cookie header is used to tie each impression to an AudienceProject panelist with if the impression is initiated from a web-environment. The individual logged entries, together with the cookie information and other information derived from additional fields such as user agent string and IP are used to generate the final AudienceReport reports.

It is important to understand that due to the unique 1:1 relation of behavioural data and panelist information in AudienceReport, we are capable of estimating not only reach in real-time, but to measure it across medias and platforms to offer unique reach across different medias and tracking-points. For a detailed description of the methodology, see [Estimation of online reach](#) and the [Getting Frequency Right](#) sections.

Discontinuation

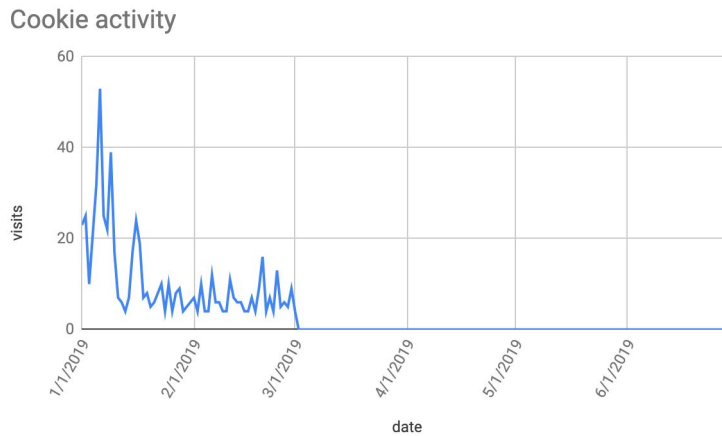
As it becomes apparent from the activation details, if a cookie stops registering traffic into the AudienceProject system for a period longer than 90 days, it is automatically phased out of the panel. The net loss of panelists is only around 4% per fortnight; meaning that 76% of a panel panellists are still active after 90 days and are active across multiple panels.

Example

UID 60ea2669-XXXX-XXXX-XXXX-240f20efe8f1 has been created in Finland, after the user completed a survey on `presentcard.fi` on the date of 2012-11-07. The user identified themselves as a 58-year old woman, with secondary school education, employed, with two children and currently living with another person, information which we encoded in the following way in the panel created on 2019-07-05:

gender	age	education	employment	income	Household size	children	weight
2	58	4	1	2	2	2	0.81

The activity pattern of this cookie in the AudienceProject ecosystem is the following:



Since the cookie was last seen on 2019-04-18, it has been part of the panel created on 2019-07-05, but phased out in the panel created on 2019-07-19.

Mobile Identifier lifecycle

Creation

Mobile identifiers are collected in two ways: directly through the UserReport mobile SDK, which developers can use to serve surveys inside their mobile application. And indirectly through AudienceReport: integrations with ad-servers (for example, Adform, AppNexus) can pass mobile identifiers using ad server macros.

Activation and contribution to measurement

In order for a mobile identifier to be considered as part of the panel, two conditions have to be met:

- The AudienceProject panel is cookie-centric and we do not operate with a separate mobile panel. This means that the mobile identifier should be directly tied to a cookie by the AudienceProject proprietary knowledge graph (AudienceGraph). AudienceProject uses graph analytics in order to determine the number of devices/identifiers that are related and belong to a specific panelist member. The ability of the graph to tie devices to panelists is nothing but astoundingly precise. When training the model against deterministic data (where we know the relationship between a panelist and device for sure), the accuracy levels average 95-96 %. For more details, visit the [Humans not devices](#) section.
- The AudienceProject system has to have seen the mobile identifier within the last 180 days on one or more of the ad campaigns or websites that we measure.

Discontinuation

As it becomes apparent from the activation details, a mobile identifier gets phased out of the panel automatically in one of the following situations:

- It stops registering traffic into the AudienceProject system for a period longer than 180 days.

