merapar

Whitepaper

Customer Data Platforms

for direct to consumer media organisations

Lead. Don't follow

Whitepaper by
Paul Stallard, Principal Technologist
Merapar

Customer Data Platforms

for direct to consumer media organisations

The key to knowing your customers and maximising your business potential

TV viewing habits are changing, driving a gradual decline in traditional TV revenues. To meet the changing user demands, bolster revenue, and maximise the return on their content investments, media companies are increasingly creating direct-to-consumer offerings.

First- generation streaming platforms were focused on presenting the content catalogue, and delivering high quality video. While those core components remain critically important, modern platforms focus on customer acquisition, engagement, and retention. All of these are driven by the timely availability of high quality data.

Customer Data Platforms (CDPs) are well known in e-commerce. Their primary role is to collect data from multiple disparate sources and aggregate the data to form a rich model of each customer, potential customer, or even former customer. We will outline some of the many ways these data models can be used to build business value, from improved user experience, to better personalisation and targeted marketing campaigns.

CDPs have proven critical in differentiating eCommerce platforms in a congested and highly competitive market. In this paper, we will explore how CDPs can provide similar benefits for media companies, as well as highlighting some key areas where media companies should focus their efforts to ensure success.

What is a CDP?

A customer data platform (CDP) is a collection of software which creates a persistent, unified customer database that is accessible to other systems. Data is pulled from multiple sources, cleaned and combined to create a single customer profile. This structured data is then made available to other marketing systems.

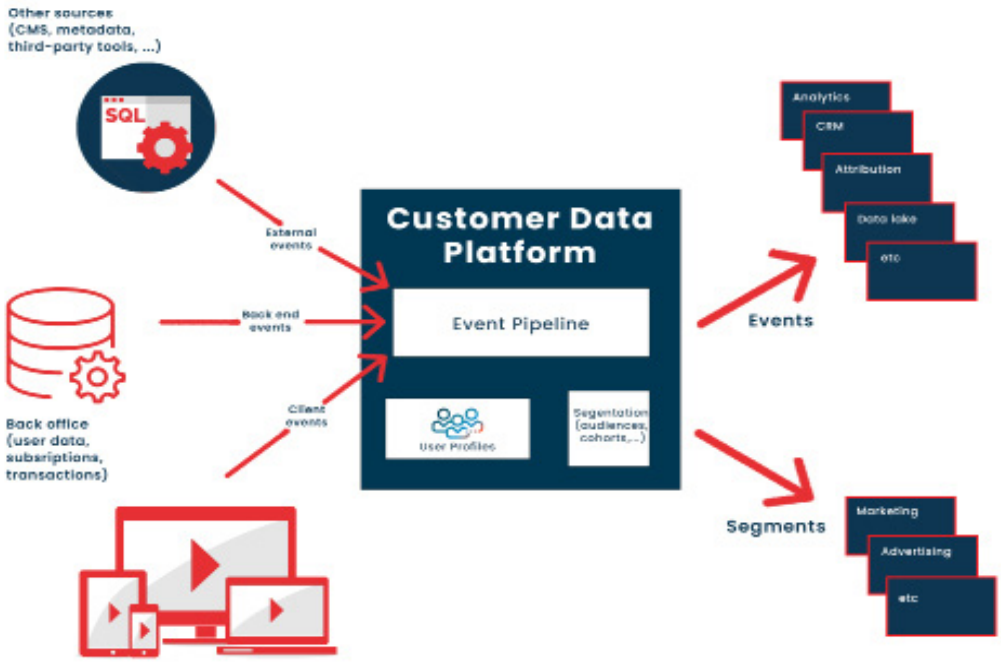
Collecting data about how users interact with a service can be complicated. Traditionally, clients are instrumented with multiple trackers, each sending some subset of data, to a set of servers dedicated to that application.

Analytics trackers may include, but are not limited to:

- sending data to an analytics platform
- marketing pixels sending data to marketing platforms
- ad trackers sending data to advertising systems
- data on video playback being sent to a Quality of Experience (QOE) aggregator

There is no central data collection, meaning that no single service has visibility and control of this data, and it's often impossible to track an individual as they move between devices.

CDPs aim to solve these problems. A CDP can be viewed as a centralised data pipeline, receiving events from clients and other systems, and creating an aggregated view of each individual customer. Events and changes to the customer view can be shared with data consumers in near real-time.



Unified Customer View

One of the CDP's primary roles is to associate each event with the correct customer profile. For users logged in to a service, this may be via a unique internal customer ID passed from the client with each event. This is much more a challenge for anonymous users, where we rely on device IDs and/or advertising IDs to infer an identity. Data from external sources might use another ID, email address for example, as the primary identity.

Which IDs are used, and the priority given to each are key decisions in the design and configuration of the CDP.

Profile Enrichment

As the CDP processes events, it builds a customer profile containing the relevant state of the user. During the design and deployment process, one of the tasks is to determine which event attributes are limited in scope to the event itself, and which constitute longer-term knowledge (state) about the user.

Segmentation

Most CDPs will provide a way to create user segments (also known as cohorts, audiences etc.) Segments are groupings of users that match a set of constraints. As a customer interacts with the service, they will move in and out of segments based on their actions.

Segments vary in complexity - an audience could be everyone who subscribes to the service, users in a certain age range, or all users who have watched a specific show in the last seven days, etc.

Segments can be used for advertising targeting, product marketing, and more.

Integration with first and third-party tools

When data is properly curated, validated and enriched, it can be forwarded to other systems. Typically the external systems - analytics, customer relationship management, advertising brokers, social media platforms, attribution systems and so on - are where the business value of the data is realised.

Without CDPs, many systems rely on direct integrations between clients and data tools. The more client platforms you have, the more complex that becomes. With a CDP there is only one integration per tool. Moreover, those integrations can benefit from the data enrichment provided by the CDP's user profile.

Commercial CDPs usually provide a wide range of off-the-shelf integrations, making it very straight-forward to integrate with the most common and popular data tools. In addition, they will provide generic outputs like Amazon S3, Google Cloud Storage, Amazon Kinesis, webhooks and so on, which can be used to integrate other tools that do not have a ready-made solution.

As we'll discuss later, centralising these integrations into one place greatly simplifies data governance, data protection, and compliance with regulations such as GDPR, LGPD and CCPA.

CDP vs CRM

Customer Data Platform (CDP) and Customer Relationship Management (CRM) are really complementary services. CDP focuses on event collection, profile management and integrations with data tools. CRM is a tool for marketers to create and execute marketing campaigns and manage communications to customers via any channels available - in-app messaging, push notifications, email, etc. A CRM can utilise data from the CDP to provide more personalised communications.

Despite the separate roles, the two functions are often confused, not least because some commercial vendors have created tools that span the two functions, and blur the distinction between them.

A CRM will normally be required - either as an integrated part of the platform or as a separate component.

CDP for Media Applications

Direct to consumer media services

It's a great time to be a media consumer. The choice of platforms has never been wider with many if not most established content companies creating direct-to-consumer streaming platforms. Coupled with services from aggregators and free services such as YouTube, consumers have a huge choice over where to spend their time and money.

Media has always been a scale business. More subscribers generate more revenue which leads to a larger budget to spend on better content, which increases subscriber numbers, and so on. For ad supported business models, more subscribers increases the value of ad slots and makes segmentation and personalised advertising viable.

As direct-to-consumer platforms mature, the success of the business is driven by four key metrics:

- Acquisition cost per customer
- Retention (ability to retain customers and reduce churn)
- Average time spent on the platform (engagement)
- Average revenue per user

In reality these KPIs are interrelated; for example, low engagement is a good predictor of churn risk. By streamlining data collection, providing a comprehensive customer profile, and offering a centralised point for integrating with other services, CDPs simplify the implementation of use cases that can drive each of these key metrics.

CDP enabled use cases

A lot of the requirements and use cases for CDP are common between applications. Whether your application is an OTT video service or an e-commerce store front, it is important to understand who is using the app, predict what they would like to do next, and provide them with relevant marketing calls-to-action (CTAs). However there are elements of video platforms that are worth special consideration and may not be handled natively by some of the off-the-shelf offerings. Some of those will be covered later in this section.

A small selection of typical use cases are shown in the table below, along with the key performance indicators (KPIs) they are designed to influence.

TYPE	USE CASE	KPI's			
		Acquisition	Retention	Engagement	Revenue
CRM	Marketing to non-subscribers	X			
CRM	Marketing to subcribers, e.g. new shows/episode notifications		X	X	
Ads	Targeted advertising				X
Ads	Ad attribution				X (cost reduction)
Analytics	Better user experience		X	X	
Personalisation	Recommendations			X	
Personalisation	A/B testing			X	
CRM	Cancellation Campaigns		X		
CRM	Win back campaigns	X			
Customer Support	Providing the suport agent with information on the user they are dealing with		X		

User experience optimisation and A/B testing

Most video applications present the user with content arranged in rows (or rails) of thumbnail images. The ordering of the rails, and the ordering of the content within each rail has a significant impact on users finding content they want quickly. That in turn impacts key metrics such as minutes watched, as well as ad revenue for ad-supported services. Should the "Continue watching" rail come before the "Chosen for you" rail? Which order should genre-based rails be presented?

There is no universally optimal ordering, so applications use personalisation to dynamically order content based on knowledge about each user's previous viewing habits. It makes sense to have the drama rail above the sports rail for someone who predominantly watches drama, and the other way around for someone who watches mainly sports. A CDP provides a hub for the data required to make these decisions.

A/B testing is a technique to determine which personalisations to deploy, and which variant of a given algorithm gives the best results. Just like a randomised medical trial, A/B testing involves exposing a random subset of the population to one algorithm (treatment) and the remainder to an alternative algorithm (treatment). The experiment runs for a period of time and an analysis of the user behaviours can determine if there's a statistically significant difference between the two treatments.

Calculating the effectiveness of different treatments, and managing the process of running multiple concurrent experiments while preserving statistical independence is normally the task of a specialist A/B testing platform. However the CDP plays a valuable role. As all the usage events are already coming to the CDP, it's the natural integration point for the A/B testing vendor. The testing platform can be forwarded the subset of events that are relevant to the experiments.

CDPs can also provide the functionality to split audiences into random subgroups. A segment can be defined and users can be randomly split into two or more sub-segments, which in turn determine which treatment a user receives. Although A/B testing is often thought of as having two equally sized groups, in reality there may be more than two treatments, and it's possible to limit a treatment to a smaller share of the total population (eg 10% of users get the new algorithm and 90% see the current version).

Data Enrichment

Traditional client-side tracking requires the client to have access to all the data required to populate the event. In many cases there is a tradeoff between providing the client with additional data that's not required for the client experience itself, and forgoing that data in analytics. Richer client-side events also have an impact on the network bandwidth requirements for the tracking traffic.

Server-side data enrichment is a significant benefit of using a CDP. With server-side event integrations, it is possible to enrich events with data from other sources before they are forwarded to other systems, simultaneously helping to provide richer data, and reduce the client burden and network overhead. For example, a client might provide a unique content identifier for a playback event, and the server-side enrichment can add title, genre, rating, etc from data taken directly from the content management system.

Video Measurement

Video measurement is one of the most fundamental, yet challenging, tasks for any video platform. There are multiple uses for the data, which may require different methods of collection, and different data points.

Video players typically generate low-level events every time something changes - a video is started, a user presses pause, a user starts to scrub, the player is forced to stall, a lower-bitrate segment is requested because of network congestion, etc. This low-level data is valuable for operations and monitoring, but less so for the CDP and the consumers of its data. Given the scale of data generated from video consumption, it is unlikely to be beneficial to send all the data to the CDP.

Tracking Video Consumption

For most analytics and CRM use cases, it is the high-level video consumption metrics that are important - what content was watched, how much of the asset was watched, did the user scrub, did they pause or skip, did they allow the auto-play feature to start the next episode, or did they interact with the video in some other way?

Unfortunately, tracking just start and stop events is misleading. Not all sessions are terminated cleanly,

Often, heartbeats are used to give a more accurate picture. These are events sent from the video player every 10-30 seconds showing the current position within the asset and the current state (play, pause, etc). By analysing the stream of heartbeats for a given viewing session, it is possible to determine exactly how the content was consumed, and in turn generate accurate higher-level events.

The heartbeat aggregation process is not normally handled by the CDP. The individual heartbeats are not independently useful so there is little value in processing them through the CDP. Instead, the high-volume heartbeat events are sent to dedicated servers, which implement the logic required to reverse engineer the session dynamics. The session metrics are then consumed by the CDP and possibly other downstream systems. Commercial platforms are available to provide this functionality.

Monitoring video quality of experience & CDN usage

Beyond consumption tracking, video platforms might also want to track quality of experience and other technical metrics of video delivery. These metrics can provide early indications of issues in the delivery chain (for example an issue with one of the CDN providers) as well as user-level diagnostics that can assist customer service agents.

These types of metrics are most valuable in real time, to ensure the system is running as intended and diagnose any issues. This is usually the domain of a specialist vendor who provides a set of client SDKs that are integrated with the major video players, and a scalable backend service to collect the data. The vendor will also provide real-time analysis and dashboarding, with the ability to define alarms that can be integrated with your existing operations centre. It is not usual for this data to be sent to the CDP.

Typical technical video monitoring might include metrics such as:

- which CDN is delivering the stream
- time to first frame
- buffering (stalls)
- average bitrate

Audience Measurement

Many countries require audience measurement data to be shared with industry bodies (often known as Joint Industry Committees, or JICs). These bodies provide an agreed aggregation of usage data, and often provide the metrics on which ad pricing is based.

JICs rely on a national-level agreement amongst broadcasters and content providers about how usage information will be tracked. This often goes down to specifying a specific data collection SDK and detailed specification of required events.

It might be unavoidable to implement the specified SDK on each client, but it is worth exploring the option of integrating server-side instead. All the necessary events may already be generated by the clients, and having one implementation of the JICs API is much more manageable than multiple client integrations. Any data issues identified in production can be resolved by server-side changes rather than requiring multiple clients to build and release new versions.

Off-line viewing

The ability to download content for off-line viewing is a popular feature of many video applications. Some users will default to this model of consumption when available, to ensure uninterrupted playback regardless of network quality. It is therefore important to track off-line viewing as precisely as on-line viewing, to ensure both aggregated viewing metrics and an individual's viewing history is correctly captured. Metrics like "total minutes watched per user" is a common KPI for video service providers, and it needs to work correctly for offline viewing.

The simplest solution to off-line tracking is to generate each event as normal, and store them locally on the device until a network connection is available. Although it's always possible for a device to run out of storage capacity after prolonged off-line use, in most cases the data generated will be approximately proportional to the length of the content. Allocating a fixed overhead when allowing a download will therefore suffice in most circumstances.

An area to consider is how stored events will be received when the device is eventually reconnected. To get the correct time of consumption, watch duration etc, the event timestamps must be respected. It may also require an adjustment to reporting processes as aggregate viewing for a given time period may change in the future as off-line devices are reconnected.

As well as the playback events, it is worth adding tracking for when a user downloads a piece of content, when downloaded content is deleted (and whether it has been watched) or when a fixed-term licence expires. These metrics will expose the user's pattern of consumption, and can help trigger personalised alerts / reminders etc.

Build or buy?

Once decided on a CDP, the next decision is whether to choose an off-the-shelf product from one of the established CDP vendors, or to go it alone and build a platform in-house. As with any build/buy decision it is worth considering some key questions before evaluating the options in more detail:

- **Do you have the technical capabilities, experience, and capacity to build a solution?**
- **What budget has been allocated?**
- **What is the timescale for deployment?**

Building your own CDP

At its core, a CDP is built around a scalable event pipeline. With modern cloud services it is relatively straightforward to build the data pipeline. There are services available for collecting high-volume events, services for processing the events (ranging from event-triggered serverless functions to managed Spark clusters) and near-limitless low-cost storage to act as a data lake. Columnar databases are available, optimised for large-scale analytics workloads.

Even with these ready-made core components, there will still be significant development required to achieve functional parity with commercial vendors.

PROS	CONS
A custom-built solution can be tailored to your exact needs.	Opportunity cost. Could the development and operations team be creating more value working elsewhere in your technology stack?
Commercial CDPs offer a lot of functionality and inevitably the cost of developing and maintaining those features is reflected in the pricing. Building your own means not paying for functionality you don't need.	To approach the functionality of commercial vendors (if needed) will require a significant commitment in resources and time. Resources will be committed indefinitely to add new capabilities and maintain the platform.
Integrations with other in-house systems are also likely to be simpler when you control both sides of the interface.	Need to implement, test and maintain each 3rd party interface. Complexity grows with the number of integrations.
Complete control over the product roadmap.	Solely responsible for the full cost of development, maintenance and operation.

Features such as identity resolution, audience segmentation, a user interface with sophisticated role-based access control (RBAC), API access to data and configuration, etc all represent significant development effort. It is worth assessing to what extent these are required and trying to remove or minimise the scope of these features to keep the project manageable.

Partnering with a CDP vendor

There are several major CDP vendors who have invested heavily in developing their platforms. They have large dedicated engineering and operations teams that are constantly adding new features and integrations.

In choosing a platform you can expect a robust set of capabilities beyond the core functionality:

- Off the shelf integrations (tested and maintained) with the most popular CRM, advertising, analytics, marketing etc vendors.
- Client libraries available for multiple platforms to simplify implementation and manage complex tasks such as identity and consent management.
- Robust identity mechanisms.

PROS	CONS
Faster time to value. The CDP will be proven and the vendor is on-hand to provide guidance with implementation.	Cost (depending on the scope used)
There will be an active roadmap, with regular feature updates. Although some new features might not be relevant to your business, others may spark ideas that would not have occurred internally.	No direct control over the roadmap. Vendors will prioritise features for anchor customers or broadly requested across their customer base, which may not include your specific requirements.
Integrations with many key vendors will be available out of the box.	Could require changes to your data model and business processes to fit with its operating model.
Access to the experience and expertise of the vendor's project team.	Exposure to the chosen vendor's long-term viability or ownership.
Development costs are shared across multiple customers.	

Choosing the buy option doesn't make deploying a CDP a simple process. Any CDP implementation will be somewhat bespoke because everyone's clients, data model and set of partners are different. There will be effort to get this working, but there will be support on hand from a vendor who will have seen many of the implementation issues play out many times before.

Implementing a robust data architecture

A robust data architecture is essential to maximise the value of any data platform. A sound, well-defined and validated data model underpins data integrity - knowing that data is reliable, correct, consistent, and compliant. Data will never be perfect, but having data that can be trusted and that can be used consistently across the business is critical to success.

Data Model

There must be one data model across all the platforms generating data. The events and attributes generated by the iOS client should be the same as those for Android, or web, or any other platform. Moreover, the events should be fired consistently at the same points in the user experience. Having this data model accurately documented is one of the most important tasks for a successful CDP deployment. Any differences across platforms introduces complexity, reduces the opportunities for automated checking, and increases the chance of errors in downstream applications.

It is tempting to generate events for every conceivable action. Whilst it is worth having some foresight for future use cases, a bloated data model adds unnecessary complexity and cost.

It is a good idea to work from the highest priority use cases and define the events required to enable those use cases. When defining an event, consider if it may be applicable to other use cases, maybe with the addition of an extra attribute or two. There is a tradeoff between making an event more generic vs adding a new more targeted one. Ideally an event should be self-explanatory from the event name - if that's not the case, consider if it should be split into separate events.

As mentioned, specifying exactly when the event should fire is equally important. Should a "video start" event be sent when the user taps the play button, when the player has initialised and requested the stream from the CDN, or when the first frame of video is presented to the user? Without a clear definition, development teams will inevitably implement things slightly differently, and it becomes more complex or impossible to consistently summarise metrics across platforms.

Data Validation

Although manual QA is likely to be necessary to test new client implementations, for example, automated data validation is key to maintaining data quality at scale.

Schemas can be used in the client implementation to ensure all generated data are compliant, and automated validation should be part of any robust data pipeline. Data that fails to meet the specification can either be discarded, or quarantined. Quarantining data makes it possible to adjust the data and resubmit it, but doing that in a timely manner may not be practicable at scale.

The CDP is likely to have the option to upload the data model and automatically validate all incoming data. If not, or if the business prefers, there are other standalone solutions for data validation such as the popular open source project Great Expectations (greatexpectations.io). Other tools such as Schema Registry can also be used.

Deprecating data elements

Data models have a tendency to grow, and with that growth comes added complexity and maintenance. Data consumers will create new use cases, especially after seeing success with those already deployed. Many new use cases won't require new data, especially if the data model is designed with some flexibility from the outset, but inevitably there will be cases when some new data is required - a new event or set of events, some new attributes added to existing events, etc. Increasing the value of the data (and therefore the CDP) is important of course so these new use cases are to be encouraged.

It is equally true that not all use cases will be successful, or that some use cases can be realised in a different way than originally anticipated. There will almost certainly be some events, and probably quite a number of event attributes, that are not used at all. This is an opportunity to prune the data model and remove the unused data points. This helps reduce storage and processing costs, reduce the complexity of new client implementations, and simplify testing.

Avoiding data model mismatch

Another potential problem comes when data is forwarded to systems with a different internal data model. Normally the model built for the CDP will be aligned with the backend business systems, but other systems (particularly off-the-shelf services) are unlikely to match. The solution may be as simple as agreeing to different terminology (is a package a saleable item containing multiple products, or is a product the saleable item that may contain one or more packages?) but it may be more complicated if the system makes assumptions that are not true for your data model.

Most of these issues can be worked around, but it is important to be aware of them and consider the implications in advance. A data model mismatch shouldn't be something you discover in production.

Feature Store

The data produced by client devices, and to some extent the back end services as well, are low-level atomic operations. The user clicked this button, they opened this page of the application, they selected to watch this piece of content, they purchased a new subscription, etc.

The use cases are often framed at a higher level and are then mapped to a specific sequence of low level events. For example, the business might be interested in the purchase funnel, measuring or predicting churn, content discovery, etc.

One option is that each team interprets the low-level events independently. The analytics team and the CRM team might independently reverse engineer the purchase funnel journey. Apart from the obvious

duplication of effort, there is a very real danger that teams interpret the data slightly differently and talk about the same concept but with a slightly different meaning. Take "churn" for example - is it reasonable to say a customer has churned when they confirm the cancellation of their subscription? Or should it only be when the remaining time on their subscription has expired and they no longer have access to subscription content?

Implementing a feature store can help both reduce overall effort, and ensure that the business is both measuring and triggering campaigns based on the same definitions. A feature store provides one place to define these higher level metrics - one consistent definition, one place to test, and one place to change should the need arise. These higher-level features can then be forwarded to downstream consumers rather than the continuent low level events.

Feature stores may be part of the CDP, or could be a separate system sitting alongside the CDP. As with many of these core data components, there are multiple popular open-source options if you decide to implement your own feature store.

Data Governance

Customer data must always be handled responsibly and in accordance with privacy and data protection regulations. As all data is centralised through the CDP, the CDP must be a special focus for data governance. The CDP can provide a centralised portal to configure data governance policies.*

** Nothing in this document should be construed as legal advice or opinion. Your own legal, privacy and data governance teams will provide the framework in which you are required to operate. The information below is based on our experience working in different legal environments.*

Regulatory Compliance

In many jurisdictions there is a legal framework for handling and processing user data. Europe has the General Data Protection Regulation (GDPR), Brazil has the Lei Geral de Proteção de Dados (LGPD), California has the California Consumer Privacy Act (CCPA), etc.

These legislations are quite different and each one that applies should be studied in detail. However, each has some basic features that could be considered good data hygiene for any environment.

Data Minimisation

All data should be collected with a purpose or purposes in mind. In some jurisdictions, users will be required to consent to their data being used for the stated purposes, and given the option to opt out.

Data minimization is the principle that any data collected must be necessary for the stated purposes for which they are processed. It is not acceptable to speculatively collect data that might be useful in the future but for which there is no current application.

Data minimization starts with the data model that should be limited to data required for the business's various data use cases. The CDP will integrate with multiple systems and each of those systems may only require a subset of the data model. The CDP should provide a mechanism to control which events, and which attributes of each event are shared with each partner.

Erasure and access requests

It may be a requirement to provide end users with the option for their data to be deleted from your systems, and for users to request a report of all the data held on them.

Erasure requests are relatively common and even when not explicitly requested, it may be required to delete data of all users who delete their account. As data is collected in order to deliver a service, it is hard to argue that the legal basis is still valid if they are no longer using the service.

Erasure requests need to propagate to all systems that hold personal data. It is well worth automating this process and the CDP is a convenient place to centralise the forwarding of erasure requests. It should

be considered part of the integration to include implementing a systems data governance API.

Although access requests are less common, it is a good idea to automate the process where possible. As the CDP receives all the customer data and only subsets of the data are forwarded to other systems, a report of the data held in the CDP may be sufficient to fulfil the access request.

Retention Policies

In line with data minimization, data should not be retained indefinitely. Some legislations give guidance on specific retention periods whereas others simply state that data should not be retained longer than necessary. Although it is tempting to hoard data forever, in reality its value inevitably decays over time. There's little value in understanding how a user interacted with an application two years ago if the application and content offering today bears little resemblance to the experience two years earlier.

Internal data governance policies might state retention periods for different types of data. These policies need to be implemented in the CDP, and also any downstream systems that receive data from the CDP.

Privacy and consent

There may be a requirement to collect granular consent from users, detailing precisely what they are willing for their data to be used for. They may choose to opt out of their data being used for analytics, or for targeted advertising, etc.

These consents will be collected on the client and sent back to the CDP. The CDP must respect those consent choices when forwarding data. If a user opts out of targeted advertising, for example, their data must not be shared with ad providers. Moreover, those user should not be included in any audiences shared with ad providers. The CDP should provide a way to configure event filters based on a user's consent choices.

Anonymisation

It is possible to retain data indefinitel and remove it from data protection legislation if it is completely anonymised.

Pseudonymisation techniques

Hashing can be used to obfuscate personal data by transforming it into a form that makes it impossible to regenerate the original value.

Hashing alone is a form of pseudonymisation and does not render the data anonymous. If, for example, a user's data is stored against a hashed representation of their email address, then anyone with the email address (and knowledge of the hash function used) will be able to recover the user's data.

GDPR, for example, encourages pseudonymisation as it makes it less likely to accidentally expose a user's data. But hashed data remains personal data under the GDPR and requires the same handling as raw personal data.

Anonymisation

As discussed, simple hashing is not sufficient to anonymise data. One possible solution is to store a random salt for each user and hash all personal data with the random salt. For as long as this salt remains stored, the data is pseudonymised.

Then, should a user request that their data be deleted, the random salt is deleted. At that point, there is no possible way to retrieve the personal data from the stored (anonymous) data, or to index the stored data using a hash of a known personal identifier. If the user ever chose to return they would be given a new salt and the data would remain separate from the old data.

The advantage with this type of anonymisation (over simply deleting all personal data fields) is that the data remains useful - it is still possible to correlate events from the same user, and metrics such as unique users, streams per user, etc remain valid.

It is worth remembering that even fully-anonymised data sets can sometimes be de-anonymised by cross referencing with another data set (so-called Linkage Attacks). It is good practice to treat anonymised data as sensitive and maintain control over its use and distribution.

Project Organisation

Agile Implementation

Implementing a CDP is a big project. There are multiple business stakeholders, each with their own requirements and use cases, multiple client platforms to be instrumented to send events, integration with back end internal systems, and multiple third party integrations requiring both technical and commercial agreements.

Trying to implement a project of this complexity on one "big bang" is likely to fail. Employing an agile methodology allows for incremental improvements, building on previous successes with manageable increments.

Create very small, achievable increments initially. Add the events SDK to the client, initialise the SDK, generate a single event, etc. In the very early stages it might be beneficial to concentrate on just one client platform. These first steps allow the development teams to gainconfidence in the process while introducing very little risk, concentrating on the key processes of identity and event generation. Once events are being generated it's possible to start configuring and testing the CDP, and building out the downstream integrations.

At this point it will be useful to identify a small core set of events that will enable multiple use cases (e.g. session start, stop, page view, playback start). Having this basic core implemented across all platforms will allow the business to start enabling use-cases and deriving value from the CDP.

Get the marketing, CRM, advertising, analytics teams involved early on to start building out use cases with the implementation. With the stakeholders using the data, requirements will often change or be reprioritized, and new use cases will be created. Don't be afraid to revisit the data plan and adapt as necessary.

Testing/QA

It can be highly valuable to have a dedicated data QA team to validate the data flow from clients, through the CDP and on to downstream integrations. Given the importance of client devices adhering to the data model and generating events at the correct point in the user experience, it is worth the investment to ensure the implementation is correct.

Existing client platform test teams often find it hard to incorporate analytics testing as it requires different tools, access to different systems, and often different skills.

The data QA team perform a range of tasks:

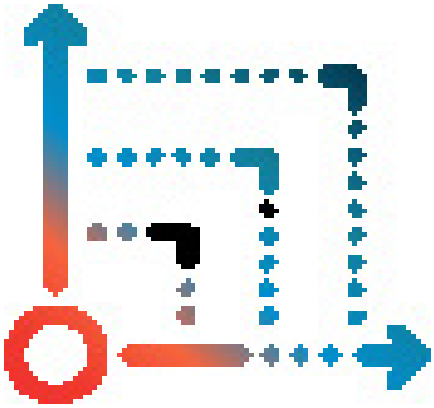
- Validation of client implementations
- Ensure correct events, fired at the right time
- Ensure consistency across clients
- Test propagation of data to downstream systems, understanding any transformations that take place as part of the integration

With multiple clients on different release schedules, plus changes to the backend integrations, it's impossible to manually QA every release of every client. Automated testing should be used where possible for regression testing, leaving the data QA team to concentrate on new product features that have a specific tracking impact.

Measuring Success

Ultimately the success of any CDP implementation will be the value driven from the use cases it enables, and the key performance indicators (KPIs) that those use cases are designed to influence. Has there been a measurable improvement in acquisition, a measurable reduction in churn, are users engaging more with the platform, are marketing spends yielding better returns, is the advertising inventory delivering more revenue?

For the deployment project itself, success should be measured in terms of time taken to reach the milestones that enable these use cases. For example, the time taken to:



- Deploy the core CDP platform?
- Make the necessary changes to the first client platform?
- Add a new client (once the process is proven)?
- Add a new event or segment integration?

Tips for Success

Implementing a CDP is a complex project but there are some common areas that we have seen often cause problems.

- **Ensure executive support**

Even though a CDP project should be able to demonstrate a healthy business case, buy-in across the organisation is important for success. In the early implementation stages, the project will be competing for client development resources with features that have a direct impact on fee-paying customers. The business leaders need to believe in the value of the CDP and ensure the implementation is appropriately prioritised.

- **Concentrate on identity and log in**

Make sure users are tracked consistently across platforms. If understanding the onboarding process is important, make sure users can be tracked before they have accounts and that their pre-registration events are part of the same user profile. This can be complicated to get right, but worth the investment in time.

- **Favour server-side complexity over client-side complexity**

This principle is well-understood by media businesses where encoder complexity has always been preferred to decoder complexity. The same is true for data integrations - any change or new functionality that is required in the client has to be implemented (and tested and maintained) for every client platform, and any bug discovered will have to wait for another client release. Always consider if it's possible to achieve the same outcome with server-side functionality.

- **Automate wherever possible**

As the solution grows (more client devices, more content, more users, more data) the complexity of maintaining the platform can become a burden. Automating client testing and data validation will greatly reduce the workload, as well as reduce the impact of any issues that find their way to production. There is a lot of configuration to maintain in a CDP - including the configuration that controls data forwarding and ultimately the compliance of your business - so look for ways to validate that configuration and implement controls to prevent accidental or unauthorised changes.

- **Avoid multiple data islands**

It is very easy to create (or inherit) a situation where different groups within the organisation maintain independent data stores with disparate goals and stakeholders. Unifying data collection into one place greatly simplifies data governance and compliance. It ensures a single definition of each data point and provides a focus for changes and enhancements of the data model.

How Merapar can help

Merapar has extensive experience and expertise developing Customer Data Platforms for media and broadcast applications.

Working closely with our clients to fully understand their needs and business objectives, we empower them to adopt change, to truly transform their products and services. We fully understand the challenges and areas where it pays for an organisation to focus their time, integrating early on in the project which is key to successful CDP implementation.

We use key concepts, principles, and approaches that are used to design systems where performance, scalability, and availability are key necessities. Our approach is always underpinned by agile practices and methodology, to ensure that projects of any size are managed to success. This agile approach is key for CDP implementation, and we work with our clients to develop in increments, allowing them to test, measure and adapt where necessary, for optimum success.

We are proud to have worked on a CDP implementation for a major D2C business. This project included the designing of a data plan, liaising with internal and external stakeholders, and providing programme management and QA capabilities. This collaboration resulted in a highly successful launch, with the business reaping instant benefits for their services. This CDP continues to grow in value and Merapar are proud to be driving this.

What Merapar customers say

Data is at the centre of our business agenda and we always encourage data-led decisions and implement a 'data-first' strategy. In doing so we spotted the need to aggregate data sources to track customer behaviour across platforms which is why we opted to implement a Customer Data Platform (CDP) allowing us to unify our online and offline data to create a Single Customer View. This implementation proved to be very successful allowing us to partner closer with the CRM and Marketing teams to drive advanced CRM, personalisation and recommendations providing a deeper understanding of our targeted audiences.

As businesses continue to grow and draw in new audiences, the customers drive the business strategy which is why unifying customer data and providing the ability to target segmented audiences becomes important. To meet these goals, many businesses have turned to CDPs (such as mParticle). We partnered with Merapar to support us with the implementation which became a flawless end to end contribution. Merapar has an incredible team of experts that work collaboratively with the business and contribute to delivering the company strategy and goals.

Seema Mandhra
VP Data & Analytics at Warner Brothers Discovery

merapar

Get in touch

Talk to Merapar if:

- **you are a media company who would like to know more about what a CDP might do for your business.**
- **you are considering implementing a CDP and would like some expert advice.**

Click here

Customer Data Platforms