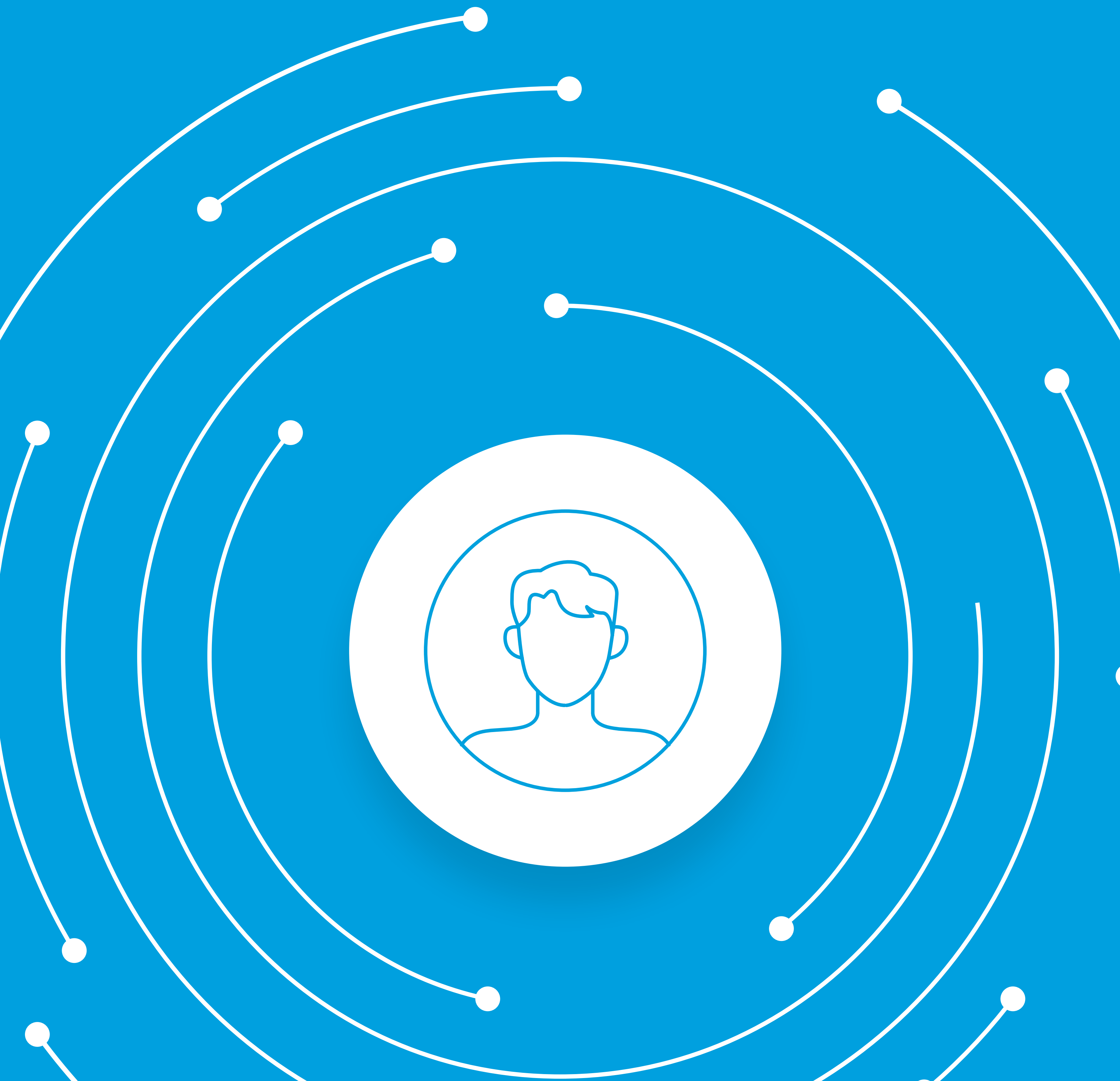
MuleSoft®

WHITEPAPER

Building a single customer view using APIs

How to create compelling customer experiences



Introduction

Data is the enterprise's most valuable, untapped resource. According to McKinsey, "Companies that create exceptional customer experiences can set themselves apart from their competitors...armed with advanced analytics, customer experience leaders achieve revenue gains of 5 to 10 percent and reduce costs by 15 to 25 percent."

Access to data — particularly customer data — is critical for companies looking to transform customer experience; they need to be able to understand who their customers are, their customers' needs, and how customers interact with the company in order to create positive experiences and stand apart from the competition.

This is why organizations start initiatives to create a 360-degree customer view ("customer 360") data layer that acts as both a common dictionary and source of truth about every customer interaction, at every touchpoint. The goal of a customer 360 is to provide an accurate, timely, and complete view of their customers so that an organization's stakeholders (i.e. employees, partners, customers, and systems) have the information they need to inform key decisions. But merely having the data isn't enough; leading companies go a step further and automatically trigger actions and business processes based on that customer information.

Fewer than 10% of companies actually have customer 360 views, because they can be difficult to set up. Challenges arise because of the large number of fragmented systems. In addition, the frequency and speed of change in those systems make the customer data time-consuming to transform, process, manage, secure, and access. When enterprises begin to construct their data apparatus, they often imagine a data warehouse or a data lake from which they can draw insights about their customers and value chain. However, trying to extract actionable insights out of data housed in a static data warehouse will ultimately be doomed because it is impossible to meet everyone's business needs with a single system.

A customer 360 should be thought of as a holistic system of systems, rather than a big pool of data that is extracted across from various systems to data warehouse. APIs can be used to group data in domains across systems (e.g. location, channel preferences) and used in various ways, such as combining data across two systems to understand how a customer moves across channels. This allows the business to shift their thinking from “what data do I have” to “what can I do with it.”

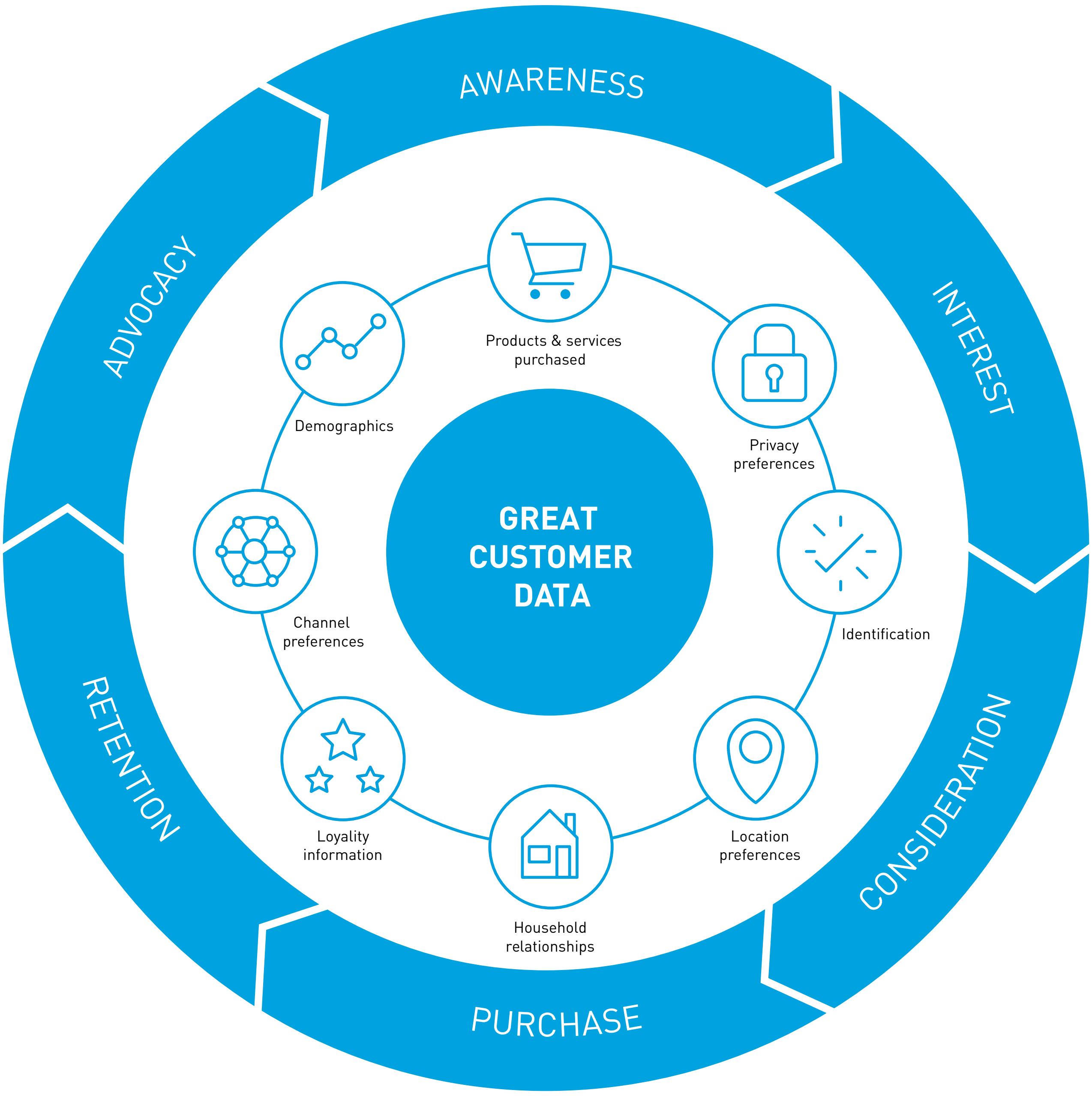
In this whitepaper, we will present an approach to modularizing your customer 360 view, to accommodate changes, trends, market shifts, and other factors in order to deliver the right experience to your customers at the right time. We will discuss an API-led approach to integrating systems to provide a well-constructed 360-degree view and we’ll also suggest tools to make this job easier. We’ll also present case studies of customers who have successfully set up customer 360 views using this approach.

What is a 360-degree view of the customer?

A 360-degree view of the customer is a *continuous discipline* that enables a company to provide an accurate, timely, and complete view of their customers so that its stakeholders (i.e. employees, partners, customers and systems) have the information they need to set the customer and company up for success.

In other words, a customer 360 doesn't stop with collecting data; a great 360-degree view of the customer is constantly refreshed and helps move the customers along the desired journey, delivering a seamless and high quality customer experience that involves a whole company, its partners, and channels.

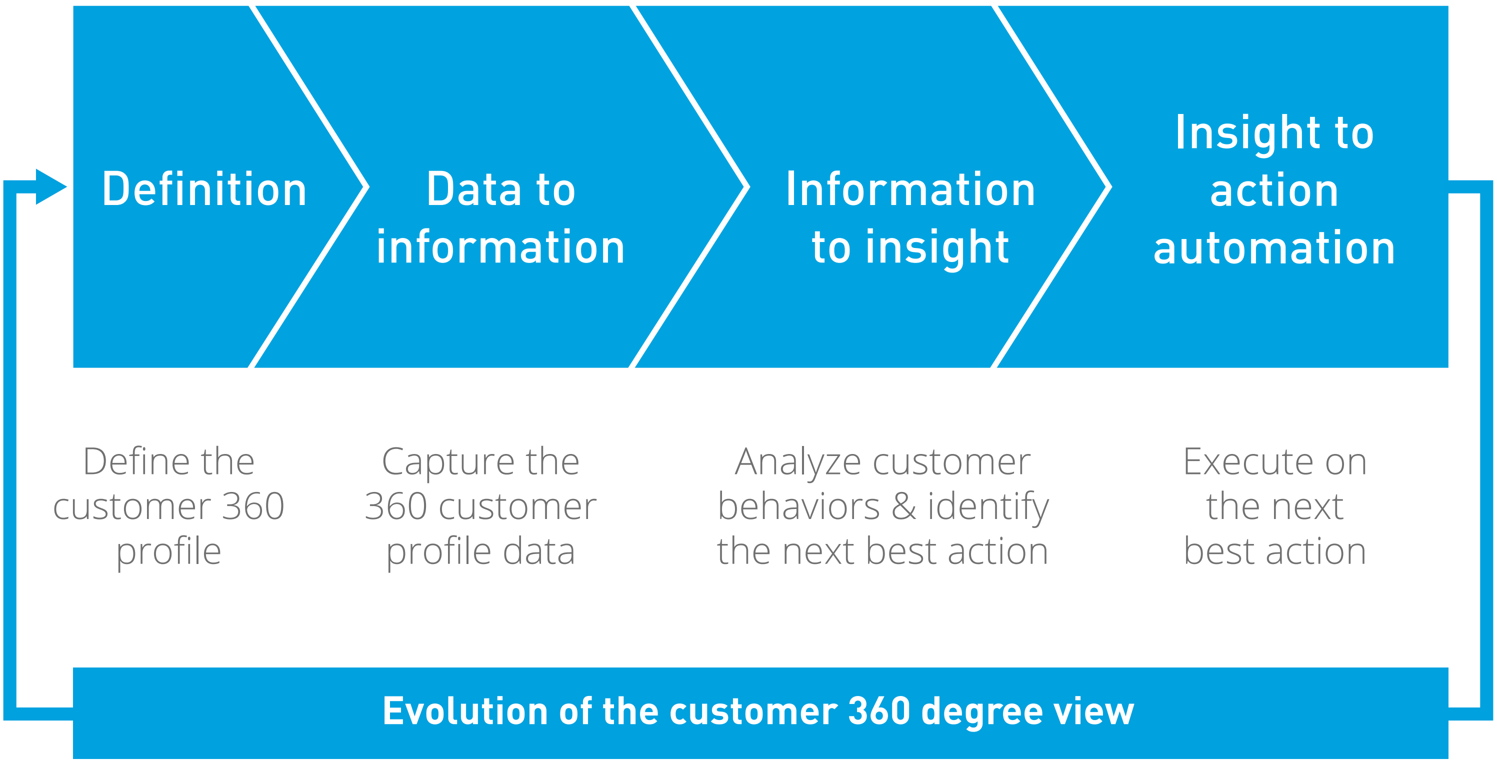
The image below illustrates common customer attributes (i.e. domains), such as location and privacy preferences, which help the business integrate their customer data in a meaningful way that can be leveraged at every point of a customer's journey.



In order to properly set up a 360-degree view of the customer, companies must meet the following requirements:

Requirements	Barrier to acheiving it
Accurate Customer data must be correct and reliable as a source of truth. In other words, the customer data must be consistent with relevant semantic definitions and standards.	Given the proliferating number of customer touchpoints, there are numerous disparate sources for customer data, and a lack of clarity around a single source of truth.
Complete Data must comprise the appropriate breadth, which usually involves first-party data (data collected from direct customer interactions), second-party data (data collected from the company's partners' interactions in any part of the value chain), and third-party data (data about the customer that is collected from external parties not directly tied to the business).	Multiple touchpoints across various channels can cause a disconnected service. Customer data comes from more than one system (e.g. in a CRM, or an access management system), and across multiple touchpoints, all of which produce different data points in ever-increasing volumes.
Timely The 360-degree view must be in real-time and reflect the most recent updates so that actions or business processes can be executed in a timely manner.	Customers now expect on-demand, proactive, and personalized service — and if they don't get it, they have no qualms about moving to your competitors.
Consistent It's necessary to apply proper, systematized governance through codified standards to ensure consistent use of the definitions and business rules that shape your 360-degree view. A critical component of governance is also consistency in security.	IT must maintain some central governance to ensure data quality and security without becoming a bottleneck to innovation. But the large number of systems containing customer information, the volume of data in inconsistent formats, and the vast number of data consumers make it difficult and time-consuming to achieve the necessary consistency.

Requirements	Barrier to acheiving it
Flexible The company continuously evolves the customer 360 as the business, tetchnologies, and customers change.	The number of systems and frequency of change in systems is a result of today's age of hyperspecialization, where what used to be done by one monolithic system now often requires dozen best-in-breed SaaS applications.
Actionable The customer 360 also contains insights on what the next best action is to acquire new customers or better serve existing ones. Integrations should feed insights to the right applications so that insights are able to be automatically executed on and thus provide practical value.	In today's hyper-specialized world, a company's ability to leverage customer data to deliver the right products and services, at the right time, and at the right place becomes a competitive differentiator. Doing this requires not only available, but also actionable data.



Definition

Data to information

Information to insight

Insight to action automation

Define the customer 360 profile

Capture the 360 customer profile data

Analyze customer behaviors & identify the next best action

Execute on the next best action

Evolution of the customer 360 degree view

Beyond these 6 requirements, a customer 360 that has true enterprise value (e.g. provides both insight and a plan for action) requires a new way to think about data. Instead of thinking about the data you have ("I must aggregate data from SFDC, SAP, and my mainframe system to build a customer 360 view"), you should think about data in domains ("I must aggregate customer, product, and billing attributes in a customer 360 view").

A domain-driven view of data offers a more complete picture of a particular customer than a more traditional view. For example, a central data management group might manage a "customer identity" domain comprised of a defined set of elements such as: name + birth date + social security number + customer ID. And the digital channels group might aggregate a different set of customer data to manage a "digital transactions" domain that consists of elements including: name, purchase history, channel purchased, chatbot history, etc.

Thinking about data across domains, rather than particular systems, allows the aggregation of data that could then be used in advantageous ways; the marketing function of that same company could have a "cross-sell opportunities" domain that combines data using elements from the previous two domains, but also relies on data sourced from third-party aggregators that use social media data. This domain-driven view, which combines numerous systems, provides a much more comprehensive picture of the customer than just the data that exists in one particular system.

This new lens breaks down system limitations by putting a new focus on business need/objectives. It forces companies to ask: who is the audience? What are they trying to do? Why does it matter? Rather than thinking about customer data from the perspective of what existing systems currently provide, organizations should identify the type of data needed to set its business up for success, then find the appropriate technologies to meet those needs.

But rethinking customer 360 as a "system of systems" – taking into account the sheer number of systems that contain customer data and the frequency with which those systems change – sounds like a herculean task to achieve. Quite frankly, most existing approaches cannot meet this demand.

Why existing approaches to customer 360 are not fit for purpose

Numerous businesses are already launching customer 360 initiatives. But in order to create the "system of systems" required to build the domain-driven view of data that companies need, constantly changing repositories of customer data must be accommodated. Here is why existing approaches to doing this generally are not effective:

- **Manual or ad-hoc processing (e.g. manual Excel extracts):**
  This often consists of manual efforts like pulling Excel extracts from each system, reviewing and updating data in each spreadsheet, then consolidating it into a single spreadsheet. This is time-consuming, error-prone, and does not ensure that the same set of data management rules are applied in a consistent fashion.
- **Syncing data across systems with point-to-point integrations built with custom code:**
  This approach involves using custom code; in other words, code that is manually developed for each system, to reconcile its data with every customer data system on an ad-hoc basis. This point-to-point approach to connectivity creates a proliferation of systems, and given the frequency of change, it becomes difficult and costly to ensure integrations remain up to date. In addition, hardcoded integrations across systems are brittle due to hard coded/inconsistent implementation patterns and poor documentation—making change management difficult. It's an inefficient way to create a robust customer 360 because it leads to duplicate processing logic across different integrations. And again, there is no way to ensure that the same set of data management rules are applied in a consistent fashion.
- **Externalizing data into a single system of record, all linked to a master reference database (e.g. hub & spoke model or an MDM approach):**
  All systems feed customer data into a single, monolithic database or system that takes on the responsibility for normalizing all data. In this instance, Master Data Management (MDM) is utilized as a middleware/integration layer and all of a company's customer data for their 360-degree view is housed in a single domain. A single system of record moves too slowly to be valuable; there are too many systems that house customer data that update too quickly to be accommodated in a single data warehouse.

It’s not uncommon for companies to start an MDM project, and then 12 months later, IT still hasn’t delivered a complete customer view that is valuable. In addition, centralization is a huge organizational commitment in terms of personnel and resources. Data stewards have to be assigned to manage definitions, cleanse data to align to those definitions, conduct audits, and enforce processes. The resulting single source of truth for all customer data leads to canonical models that are massive and hard to manage for central IT/data management.

If I have an MDM, do I need MuleSoft?

When we talk about Master Data Management (MDM), it’s important to define what that means. MDM can refer to a discipline or set of rules and processes that allow you to map and connect data from one system to another at a logical level. An example might be ensuring a Customer ID in Salesforce is the same as it is in Workday, or that birthdays are always displayed in the format DD/MM/YYYY. Sometimes, MDM also captures and records those rules as a kind of map to the rules and processes. But what has happened over time is that MDM systems have taken the role of storing the data and doing some of the mapping and overlaying the data. That is a bit more problematic.

When an MDM functions as a data lake or warehouse or data store, efforts to use it to fuel a customer 360 are doomed; systems and customer preferences are changing too quickly to provide the timely, accurate, and flexible capabilities of a valuable customer 360. As a discipline, MDM works well, and that discipline can be managed either with MuleSoft or an MDM vendor. But using MDM as a data store in itself is not recommended; we suggest you replace it with the API-led approach we propose.

What capabilities are needed for a successful customer 360?

As we've seen, the existing approaches to setting up a customer 360 - from manual spreadsheets to an MDM system - cannot create the timely, flexible, consistent, and actionable full view of customers that businesses need. These approaches simply are unable to create the continuous discipline of a customer 360 view that the multiplicity of channels and touchpoints available to customers creates. The next question to address is what approach will be sufficient? To determine that, we need to understand what capabilities an organization needs to create a successful customer 360. There are five:

- **Connectivity**
  All relevant data sources (including third-party providers) and services, such as MDM, access management, and analytics must be collected and subscribed to. Merely creating an MDM is not enough; while MDM helps create a view of the customer by providing linking and identification mechanisms, it doesn't have the breadth of connectivity to various systems that modern enterprises require. In addition, trying to integrate an MDM system with the necessary systems and applications needed for a customer 360 propagates point-to-point connectivity, which, as we have seen, is insufficient and a drain on time and resources. For a complete view, systems like identity and authorization tools, tag management, and CRM need to be incorporated as well.
- **Real-time delivery**
  A key customer 360 capability should be velocity. Data delivery should be event-based and performant enough to support high payloads, and should include visibility into how data is processed and where they breaks are when it's not. It should also not end at data delivery, the right processes can and should also be triggered that actually make a positive impact to customer interactions at the right time and place. The actions you choose to automate then become a representation of what your business philosophically wants to be—e.g. do you want to call a customer after a bad tweet?

- **Balanced governance (i.e. secure by design):**
  A central infosec team should be able to apply central management and data standards as well as security around who can access customer data, what data they can access, and drive governance without compromising business needs. Trying to take on too much of this centrally will not provide the timeliness a good customer 360 must have, but giving lines of business complete autonomy with no central governance does not work either. A balance must be struck.

- **Sensible change management.**
  Data sources and applications change often. Making those changes happen without downstream effects is essential to a well-set up customer 360. Decoupling applications and data with APIs to easily swap or add new systems, regardless of whether they're in the cloud or on-premises, is a great approach. In addition, the customer 360 view must be deployed on a platform that supports hybrid models if they run any systems on-premises.

- **Operational alignment.**
  The customer 360 program is a discipline that needs to be operationalized. Access to customer data should also be scaled and consumers of the data should know to check in to a central library first before building new code. A platform should be able to support that and enable business partners to incorporate what they find centrally into their own customer 360 efforts. Processes should also drive new data assets to be published upon implementation, so they are discoverable and consumable by others. This avoids siloed thinking by providing visibility and access to standardized data assets that drive accuracy, timeliness, and consistency.

Achieving a customer 360 with an API-led integration strategy

As we've seen, the primary pain point for organizations is the proliferation of systems and frequency of change in and of those systems, which both make it very difficult to maintain an accurate customer 360 view that is dependent on a single system of record.

The domain-driven lens on data breaks down system limitations by putting a new focus on business needs and objectives. Once businesses are ready to think about data in a domain-centric way, they need the tools to access it. Enter the modern API.

Why APIs?

APIs have emerged as a simple way to enable others to have access to data. APIs enable controlled access to a defined scope of data—free of limitations on where the data is (e.g. cloud or on-premise, which network, etc.) or exposure to the complexities of the system that houses the data. APIs have been around for decades, but recent improvements on how APIs are designed and productized have made them ideal for sharing and orchestrating data across systems:

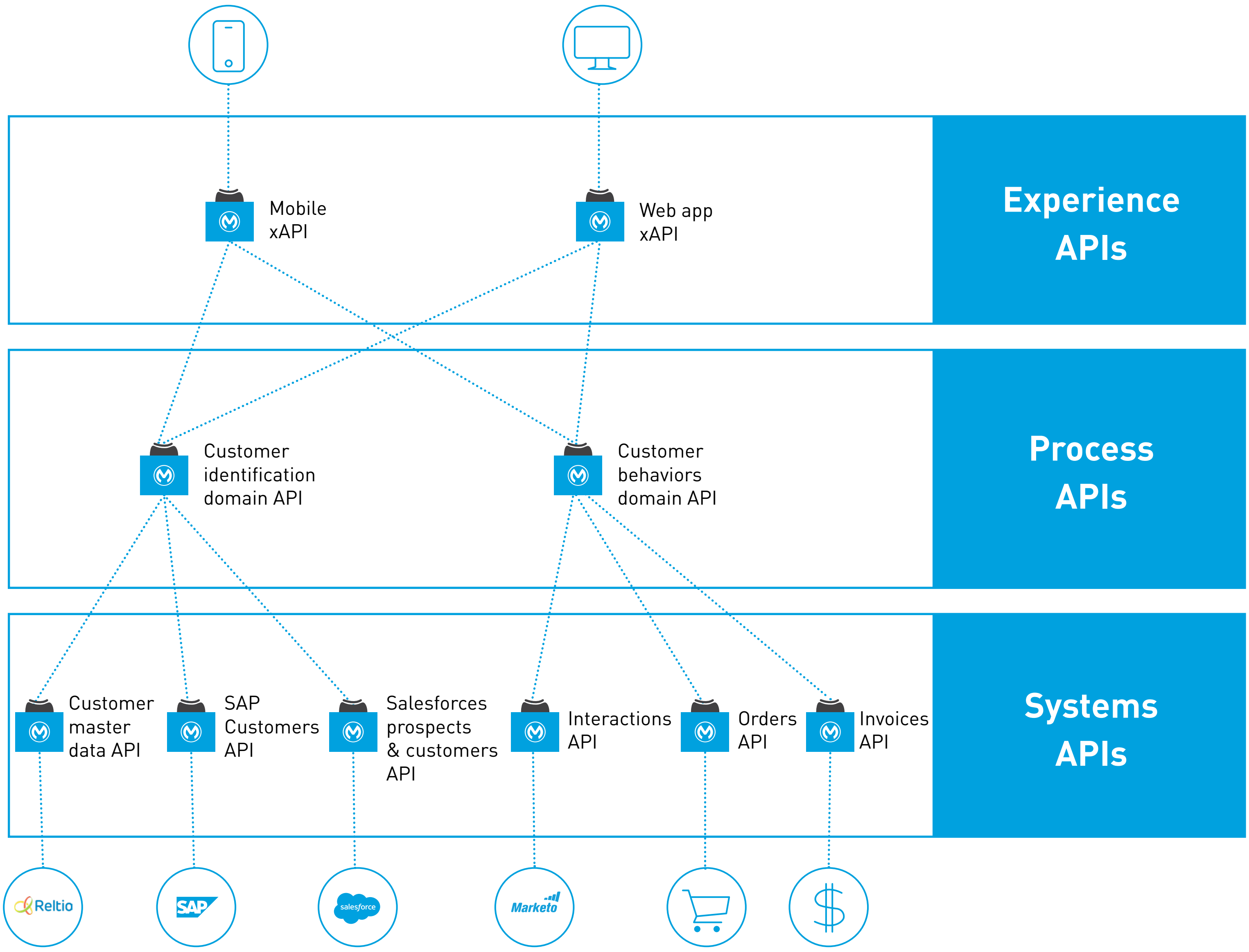
- They are based on industry standards.
- Discoverable and accessible.
- Easy to build, manage, and consume.

MOUNT SINAI

Mount Sinai is one of the oldest integrated healthcare systems in the U.S., with 7 hospitals, 1 medical school, 15 institutes, and over 40,000 employees, physicians, and residents. However, whenever a doctor saw a patient, they had to look through an average of 5 applications to get a comprehensive view of the patient's medical record. The number of siloed data sources challenged efforts to improve the patient experience and streamline how doctors provide care, demanding a new approach to connectivity. To address the lack of data interoperability, Mount Sinai turned to their IT team to transform care by building a single patient view and improving coordination between physicians, caseworkers, and community care providers.

This required breaking the silos between the medical and non-medical data, applications, systems, and devices used by Mount Sinai teams and community partners. With Anypoint Platform, Mount Sinai exposed data from systems and applications through Fast Healthcare Interoperability Resources (FHIR) APIs. The hospital system can now share data through these APIs with hundreds of community care organizations and healthcare providers across New York City; thereby improving collaboration.

An API-led approach to integration is a methodical way to connect data to applications through reusable and purposeful APIs. These APIs are developed to play a specific role—unlocking data from systems, composing data into processes, or delivering an experience. In the customer 360 context, APIs are used to organize and orchestrate data across domains to create the full 360 view. Here's how this works in practice:



System APIs expose data across business applications or data stores, such as SaaS, CRM, or legacy databases. Exposing this data through APIs enables the shared consumption of data throughout the enterprise.

Process APIs consume and orchestrate data exposed by System APIs, and represent common business processes that interact with and shape data. They exist independently of the source systems from which that data originates as well as the target channels through which that data is delivered. In this scenario, different process APIs can be used for multiple purposes including:

- Validating with master data (e.g. who is the customer)
- Querying data: e.g., grabbing aggregates, roll up data across multiple sources (data from those sources made available via system APIs per source). Multiple process APIs can be used in a customer 360 scenario to pull different views appropriately for each consumer.
- Executing actions: Factors that affect change such as:
  - Triggering an action by one system to another system to move along a business process,
  - Migrating customer data (which may also be transformed along the way) from one business application to a new target (e.g. another business application or a data store) that has been named the system of record for a domain,
  - Transforming data to align to central standards (i.e. for data governance).

The process API becomes a system of systems — comprised of a customer data layer and customer engagement hub — to abstract data from source systems, and validate, link, enrich, transform it, as well as apply business logic, and orchestrate business processes.

Experience APIs transform data and services so that they can be easily consumed by its intended audience, all from a common data source. Experience APIs can filter certain data out for different consumers and/or extend different presentation formats, which vary by the consuming device or channel.

This domain-driven approach, created with API-led connectivity, has a number of benefits:

- **Accuracy:**
  The unified platform provides end-to-end visibility across the network and processing performance, allowing users to precisely and take action swiftly to maintain the accuracy of customer data on a real-time basis.
- **Completeness:**
  Anypoint Platform provides connectivity across all systems, no matter where they are deployed, and offers detailed analytics and insights into which systems source, transform and consume data in a single management console.

- **Timeliness:**
  Users experience, on average, 60% time saved through out-of-the-box connectors and easy-to use, user-centric features such as graphical data management capabilities. There are also multiple options to quickly and efficiently design APIs to transform either standard or custom transformers with an intuitive, easy-to-use user interface.
- **Consistency:**
  MuleSoft's API management tools allow either out-of-the-box security policies or give you the option create custom policies globally, by resource, or for each node. This creates a single fabric for network-wide consistency no matter where your applications are deployed. API tiers provide multiple levels of governance and control to ensure that neither your data nor its quality are compromised. Tokenization, data encryption, and edge protection are also provided for added security, and you can control of business logic and security policies through a single management pane.
- **Actionability:**
  Anypoint Platform's built-in marketplace for reusable assets, Anypoint Exchange, offers a bottoms-up approach to data governance where lines of businesses seek and consume standardized data assets. This drives a culture of collaboration and self-service that simultaneously operationalizes your customer 360, aligns to data standards, and speeds up development.
- **Flexibility:**
  Anypoint Platform offers support for any deployment model and an easily containerized hybrid platform that provides a low-risk option to migrate to the cloud. It has a lean Runtime that supports any deployment model and is highly performant. Anypoint Platform allows simple containerization, enabling integrations to be written anywhere and moved anywhere. MuleSoft also makes it easy to store and distribute your data from and to your MDM, and not only in an point-to-point batch manner, but in a decoupled, clearly encapsulated way, allowing for reusability and scalability of your integration architecture.

Customer spotlight

An international retailer, faced with stiff competition from Amazon as well as big-box stores like Walmart and Target, decided their competitive differentiator was an emotionally resonant, personalized experience at every touchpoint in the customer journey. They believe creating this experience is the only way they can win.

They began their transformation in 2016 and did a great deal of work in the customer space.

For this retailer, their key question is "who is our customer?" How can they get data from their customer and make sure they have the right data to create the right experiences?

Initially, they tried to connect the disconnected experience across multiple channels by syncing data across all the databases where customer data resided. It was challenging, but the project was done. They started using batch-oriented processes to sync data, then they moved over to near real-time, or even real-time, but ultimately that was not the right approach. The syncing rules were very complex. Most of the time they were breaking. The customer wasn't defined in the right way for those different interactions, and at the end, the experience was not the best for the customer.

So to fix the problem, the first thing they did was to enable their integration platform. The second step was to rethink and redesign the whole customer data foundation. They will have a 360-degree view of customer profile, and customer interactions, and then expose that to those channels through a MuleSoft API, or a set of MuleSoft APIs on top of it.

Once the retailer created their customer data foundation layer, they implemented a new tool within the MDM space, and deployed a new data cleansing engine to have the right data for their customers. On top of it they placed a set of MuleSoft APIs, all-cloud based, running in Anypoint Platform, to expose the data to all the channels.

One of the APIs created was the customer profile API. This creates a profile account or transaction accounts on the .com or mobile app, will have pre-populated information when customers look at their profile. APIs also allow single-sign on to all channels, a booking tool for appointments, and special data security — applied via API policies — for the care team.

The retailer maintains separate data stores for customer profile data and MuleSoft was used to create the API on top of that data store. Then, applications can access the customer profile API for their website, mobile app, and the online reservation system which all leverage the same API and the same data store.

In addition, Anypoint Platform manages the integration strategy used with these APIs, eliminating point-to-point integration and allowing the IT team to deliver projects faster and more reliably. In short, no matter where the customer interacts with this retailer, he or she will have the same experience every time and the retailer should understand who the customer is, thanks to Anypoint Platform's provision of well-designed, well-managed APIs at the cornerstone of a holistic integration strategy.

Anypoint Platform's capabilities extend across multiple needs— API management, Message Broker, ETL, etc. All development can be done within the same IDE using Anypoint Studio with a seamless integration to Anypoint Platform in the backend. There is a reduced need for 'coding' as MuleSoft provides a WYSIWYG capability for flow and transformations implementation, saving time and resources. And MuleSoft is based on a performant and reliable cloud-based platform, providing HA capabilities and average response times below 200ms for medium-high complexity API transactions.

By using Anypoint Platform for their customer 360, the retailer enjoyed the following benefits:

- **Reusability.**
  A microservices approach, managed with Anypoint Platform, enables reusing smaller components across multiple applications, reducing development effort by approximately 50%.
- **Maintenance.**
  It's easier and less expensive to maintain smaller reusable assets rather than large complex components.

- **Time-to-Market.**
  Development time has been reduced; the retailer is building API integrations in a two-week-long sprint, instead of 6-8 weeks previously

The Senior Manager of Development of this retailer said, “MuleSoft has helped us to build those relationships with customers; in the past there was a disconnected experience across channels. Today, through MuleSoft APIs that we’re leveraging, we can provide a seamless experience so our customers know that we care about them.”

About MuleSoft

MuleSoft makes it easy to connect the world’s applications, data, and devices. With our market-leading Anypoint Platform™, companies are building application networks to fundamentally change the pace of innovation. MuleSoft’s API-led approach to connectivity gives companies new ways to reach their customers, employees, and partners. Organizations in more than 60 countries, from emerging companies to Global 500 corporations, use MuleSoft to transform their businesses.

MuleSoft®