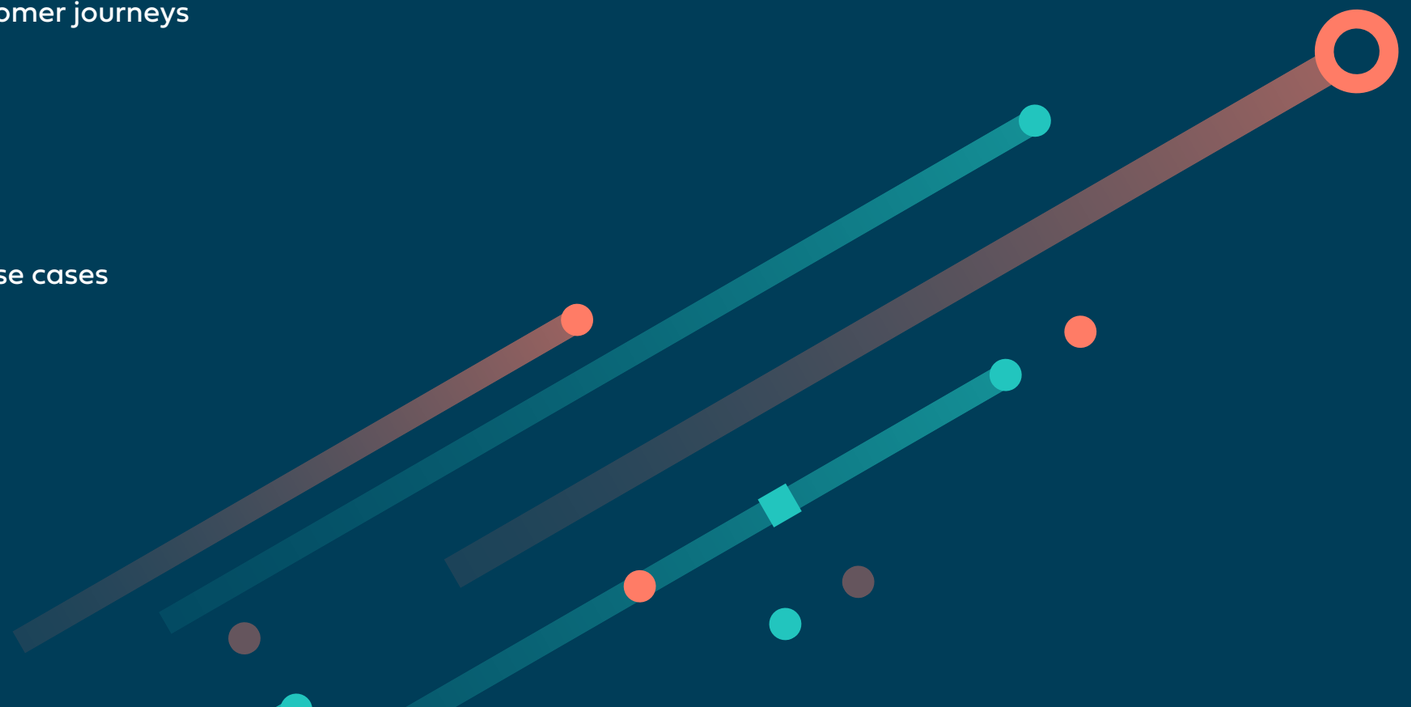
How to build a Data-Centric architecture with Boomi ?

Synchronise, qualify and activate your data
at the heart of your business

Arhis
Going beyond data

boomi

Table of Contents



Why a Data-Centric Architecture ?

It is generally agreed that data is the new black gold for businesses: it feeds all processes and exchanges. Much of this data is shared, and its governance is becoming a key requirement.

However, overcoming the siloing of business applications must also be synonymous with added business value.

The expression 'data-driven' symbolises this desire for data to drive the business. The way in which this is achieved is sometimes less obvious. At Arhis, we are working on data-centricity, or how to put data at the centre, as the foundation and at the service of business strategies.

This ambition requires one or more Data platforms associated with data exchange and processing mechanisms.

In this white paper, we will describe the various components of this organisation.
Let's summarise them:

- Accessing application data.
- Transforming data to make it operationnal.
- Build a metadata layer specific to each requirement.
- Serve business use-cases: by exposing data, delivering it to business applications or building specific functionalities from it.

To achieve this, Boomi offers a number of levers that act as accelerators, in particular through its exchange and governance platform capabilities.

A few design questions before getting started

1. Define use-cases first ?

It is common practice to start by identifying use cases. In particular, this exercise enables you to measure the most penalising pain points and draw up a backlog of priorities.

Be careful, however, not to fall into the trap of simply looking in the rear-view mirror. A data-driven strategy must enable us to open up to other and, at the very least, other ways of doing things. In practice, the most frequent conclusion to be drawn from this 'tour de piste' is to see just how diverse the diversity of use cases. Given this observation, the conclusion is to move towards an architecture that can address the widest possible set of data possible, and to be able to on board as part of a shared technology approach. Beware of platforms oriented jobs, such as marketing, because much of their underlying data belong to a common business base.

Ultimately, the approach to getting started should focus on the choice of priority areas, which most often depend on the company's business. It is obvious, for example, that in distribution, the customer and the product are the 2 strongest levers.

A few design questions before getting started

2. What architectural choices are made ?

Before starting a project, it is essential to define technical design choices that are relevant in the long term. Let's look at a few some of the most common questions:

TEMPORALITY :

Before starting a project, it is essential to define technical design choices that are relevant in the long term. Let's look at a few some of the most common questions:

- For internal flows : Event messaging
- For data exposure : API

At the target, it is best to rely on a single exchange platform that manages all the flows, usually via one or more data hubs. We'll look in more detail below at how the Boomi platform provides a solution to this challenge.

PERSISTING DATA OR NOT ?

When modelling is a key business issue, particularly for reference data, the option of persisting the data in the platform. However, 3 other complementary scenarios are possible:

- Collect/distribute data as required via dedicated connectors, without persistence (flat-passing)
- Store in databases dedicated to a particular domain, outside the company's centralised system
- Use data virtualisation to create dedicated environments

It is by studying the challenges associated with this data that the best choices can be made.
This is particularly true of transactional data, given its high volume and volatility, its semi- or unstructured nature and the diversity of storage formats.

As for the Boomi platform, it offers to manage data integration for each of these scenarios.

CLOUD ARCHITECTURE OR NOT ?

In the context of centralised data management, the use of the Cloud and in particular SaaS tools is a strong asset. We will describe below how this advantage can be used for data recovery. More generally, what we are talking about here is the ability to provision each depending on the stage of the project and potentially in run mode according to periodicity.

As an IPaaS, the Boomi platform connects these different types of instances together and natively offers the necessary scalability.

A few design questions before getting started

3. Which resources ?

The success of a strategy depends to a large extent on mobilising the right resources. 2 questions appear :

IN-HOUSE SKILLS OR INTREGRATORS ?

It is advisable to rely as much as possible on 'standards' for each component. This is a guarantee of longevity and the availability of a community of players. However, the 'tool' dimension is not the real issue in this type of project. A trusted partner should be chosen on the basis of its experience in mastering the entire value chain, in all its aspects: functional, integration, architecture, etc. This is not just a matter of scoping, the challenge is to mastering the business challenges, handling the many management rules, knowing where to do what to meet needs and optimise needs and optimise performance, not forgetting not forgetting the need to support change that optimises TCO.

TECHNICAL OR BUSINESS TEAMS ?

Too many projects fail because of apprehension between IT and business experts. The real answer is to mobilise according to their skills. To achieve this, there are no standard principles to the current organisation, combined with a future roles and processes. This is the the real content of governance.

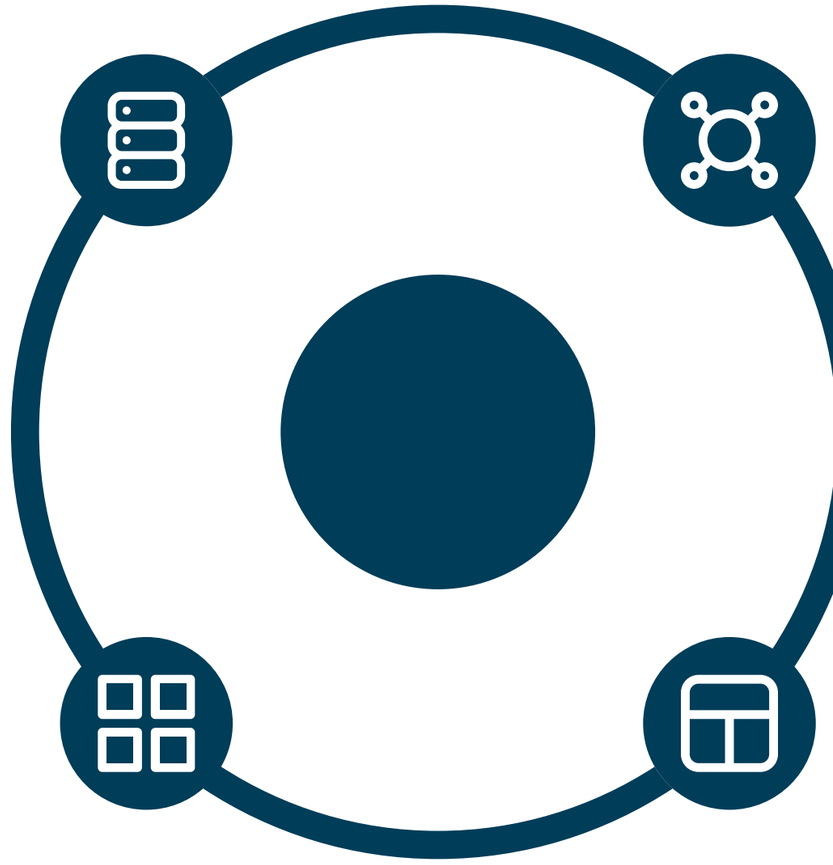
The main stages in a
Data-Centric strategy

The aim here is not to repeat the stages of a project, but to show the different facets of what can be deployed when data is brought together at the heart of the ecosystem. Depending on the existing situation and the company's priorities, these different elements will be combined in a progressive approach, which must nevertheless be sufficiently encompassing to address the first use cases from the very first iterations.

It is important to distinguish between domains and use cases. **The domains:** Customers, products, assets, etc. are each partly common to various business activities within the company. For example, a customer is the central subject for marketing, but is also a key element for the finance, logistics, compliance and other departments. On the other hand, a **use case** is linked to a particular business or department. For example, campaign management is managed by marketing. A domain's data needs to be shared and appropriated by all, whereas a use case requires indicators and functions, or even specific tools.

To get to the use case, we need to consider several 'layers' in succession:

- A platform approach in the singular or plural which serves as a foundation, and whose purpose is to make the data 'usable' for everyone.
- The activation of data from this platform, i.e. its use by business or digital application modules.
- And finally, specific systems that operate data according to layers of indicators, operations and business functionalities: in vendor solutions or by developing development of proprietary applications.



Stage 1 :
The DATA platform (s) at the heart of the ecosystem

1.1. DEFINE A PERIMETER AND ITS CONNECTIONS

For each area, there are 3 different different scopes, to be analysed and validated for each each company:

- This is the core of the target, and can be modelled within the BOM (Business Object Model). This is often referred to as the company's reference data.
- What falls within a scope shared by certain players, and which must be made available for certain scenarios, particularly transactional ones.
- What falls within a perimeter that is specific to certain which does not need to be shared. shared.

To create each of these 3 perimeters, we will integration processes will be implemented with different docking points, which may or may not include management rules, and if necessary according to different timeframes.

The opportunity of a Boomi type platform already appears to be a decisive factor, offering integration 'Anything, Anytime, Anywhere' integration. As an IPaaS (Integration Platform as a Service), the AtomSphere Boomi platform can be used to integrate any type of application regardless of the infrastructure chosen Cloud or On-Premise. As the platform is cloud-native", there's no need to install heavy client software or advanced coding skills, because Boomi has the advantage of being a low-code tool for developing complex complex integration processes.

As an iPaas solution, it supports real-time integration and flexible scaling to meet the data volume requirements requirements of any type of ecosystem.

Stage 1 :
The DATA platform (s) at the heart of the ecosystem

1.2. QUALITY AND GOVERNANCE OF DATA

As mentioned in the introduction, the aim of the data-centric platform is to enable you to work with 'usable' data.
Usable means first and foremost accurate, accessible and up to date, in compliance with regulatory regulatory constraints.

Initially, analytical and design analysis and design skills. These include in particular:

- Data profiling, to analyse existing objects and the data quality of these objects in the source applications.
- Construction of a logical model for a given perimeter, if required.
- The construction of a logical on a given perimeter.
- Building identifiers for each object and their link with those of each application.

Then the implementation phase has to deal with the data, which must be :

- Understandable and documented.
  - Objects are classified and catalogued. The objects and attributes are typed, described and associated with the business (glossary, internal wiki, etc.).
  - APIs are described by their interface interface
- Combined with quality indicators : completeness rate, distribution of compliance with expected formats, etc.
- Tracked throughout their lifecycle: applications in charge of applications in charge of collecting transforming them and the applications the applications that consume them.
- Secure access and compliance with regulatory requirements.
  - Connection to the platform or API via identification via the company directory.
  - Read/write access based on user authorisations.
  - Anonymisation of personal data (PII) according to profiles.

Then the implementation phase has to deal with the data, which must be :

- Description of objects (data catalog) and data lineage enhanced by the platform's own AI engine engine that speeds up the creation of links.
- Link to business model (business glossary)
- Anonymisation of personal data (data masking).
- Description of interface contracts and access control (API portal).

Stage 1 :
The DATA platform (s) at the heart of the ecosystem

1.3. MDM, RCU

Depending on the company's business, the modelling can be translated into an MDM for a more or less important part of the data (or RCU for the Customer domain).

It involves modelling, de-duplicating, synchronising, cleaning and enriching data from different domains and, where necessary, linking them between them.

With Boomi Master Data Hub, you can create this MDM Repository for :

- Design a model that represents the entities in each domain.
- Deploying the model and identifying the sources feed it or consume it.
- Synchronise data with Boomi Integration to orchestrate data flows and data quality.
- Governing access to functionalities made available to users.
- Validate data by resolving duplicates, correcting errors.

Finally, we construct a point of truth which constitutes the reference for the data to be distributed throughout the IS.

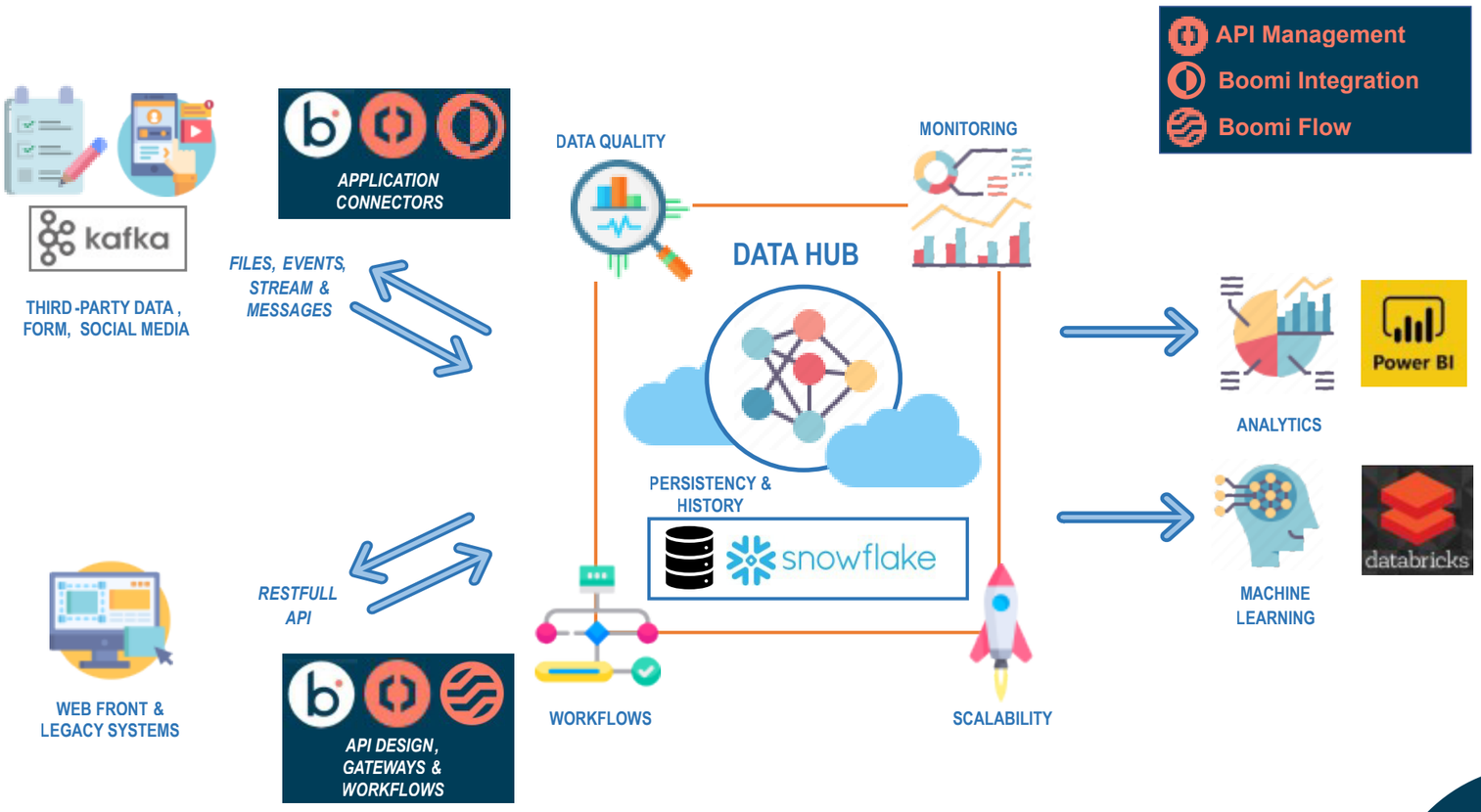
Stage 2 :

Data activation within the IT ecosystem

There are an increasing number of strategies for using data to serve the business ecosystem. While it is necessary to work on a case-by-case basis, we can give an overview of the different approaches :

- Data is synchronised as it happens with the applications that applications.
- The data is exposed in services which are used in particular by the digital front-ends.
- The data is fed into devices which order it for analysis or to display specific views.

Example of a Data platform and activation schemes.



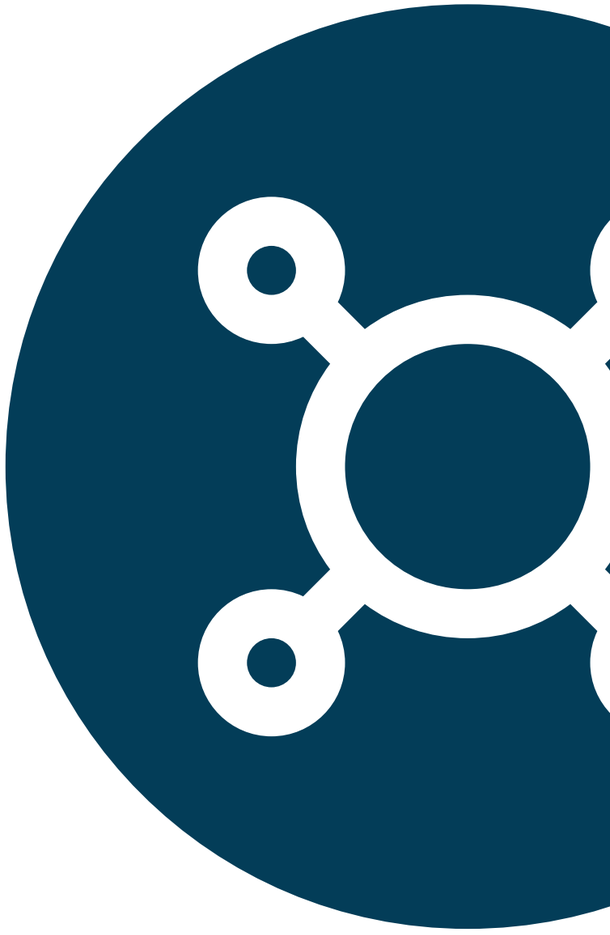
Stage 2 :

Data activation within the IT ecosystem

2.1. THE ADVANTAGES OF A TRADING PLATFORM

To be able to implement different data activation strategies, mastering a single platform is a guarantee of flexibility and efficiency, while minimising the budget. This is what Boomi offers through a range of exchange functionalities and services that services that enable shared implementation, monitoring and maintenance of connectivity. Here is a list of the main features:

- Be able to connect applications with heterogeneous interfaces using interoperable standards or application connectors.
- Manage the service lifecycle by providing an interface for creating, publishing maintain and decommission an API.
- Ensure the mediation and routing of by offering tools for transformation, rules management and validation tools.
- Orchestrate services and aggregate responses to enrich the data in an exchange.
- Monitoring the platform, tracking processes, processing and tracking of incoming and outgoing messages.
- Pool access and manage the security policy of applications accessing the platform.
- Manage versions to ensure a switchover without interrupting service continuity.



Stage 2 :

Data activation within the IT ecosystem

2.2. PROPAGATING DATA FOR CONSUMER APPLICATIONS

The most traditional way of taking advantage of the platform is to propagate data. For example, in the case of marketing, updates will be exchanged with the CRM for customer relationship management and consent, as well as for the marketing campaign management tool. As already mentioned, the best practice is to is to generate all the flows in real time. This meets the reactivity requirements of many use cases, and is facilitated by existing tools, particularly Boomi.

The two modules, Boomi Integration and API Management modules make it possible to implement using different flow patterns synchronous and asynchronous flow patterns, making integration with applications by adapting to their their interfaces.

SYNCHRONOUS PATTERN OR CONNECTED MODE

This pattern is used in real time applications where data is fetched on demand when it is needed. The application issuing the request, called the Client, uses the API to request the destination application, which will carry out the desired processing and return the response.

Constraint: All applications must be available to circulate the data, which requires requires the implementation of highly available highly available architectures.

ASYNCHRONOUS PATTERN OR OFFLINE MODE

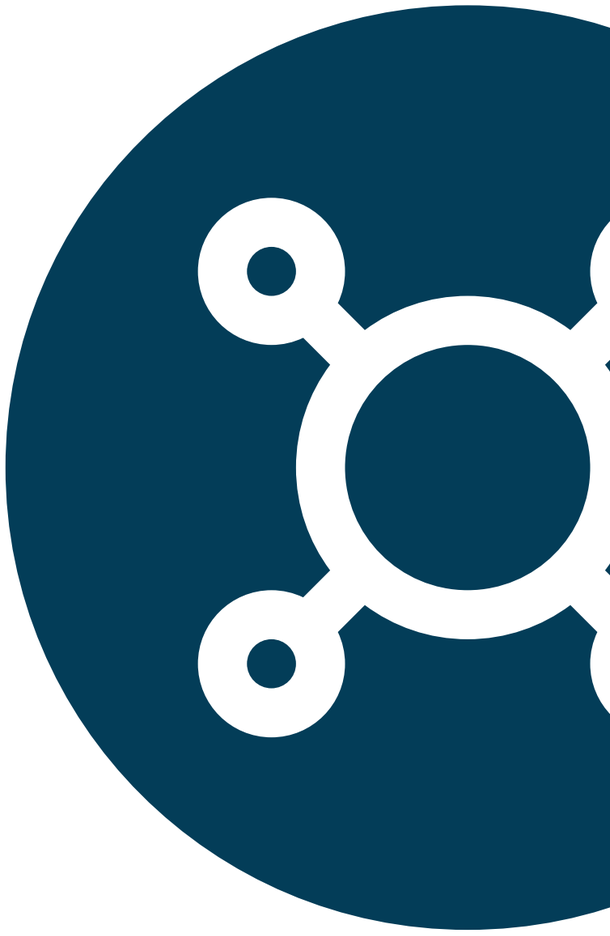
The asynchronous pattern is implemented by interacting with a MOM (Message Oriented Middleware) or Message Broker to ensure data exchange.

All the players do not need to be connected at the same time, except for the Message Broker, which must be highly available. This pattern reduces the architecture's availability constraints and cuts costs. This is why it will be favoured for internal internal exchanges.

There are 2 possibilities for implementing an asynchronous exchange :

- **Publish-suscribe:** the application producing the data publishes an event/message and the consuming applications subscribe to this type of event/message and consume them according to their needs.
- **Streaming:** the application that produces the data application publishes the events/messages events/messages in a log and the consuming applications can access access the log and join the event event stream at any time and from any anywhere they need to.

NB: The API can be used in asynchronous mode to access data from an application which does not have a connector to the MOMs. The Boomi platform will then be responsible for transforming this call into a message.



Stage 2 :

Data activation within the IT ecosystem

2.3. EXPOSING DATA IN SERVICES AND EVEN WITHIN CUSTOMER JOURNEYS

Data-centricity must also make it possible to manage interactions between the company and the outside world. The management of Boomi's APIs within a service logic is the lever that supports this strategy.

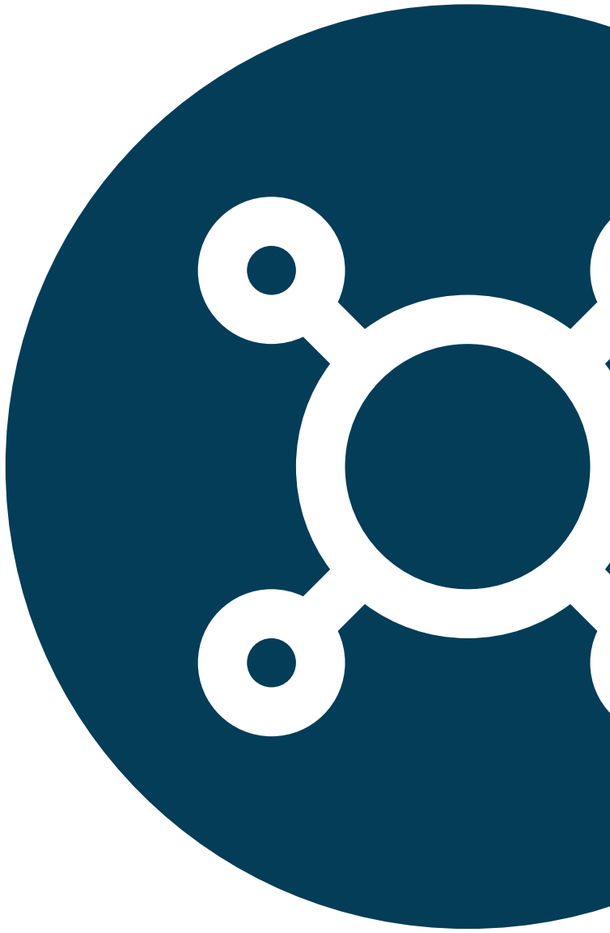
Boomi Flow is a low-code development environment for building customer journeys and automating workflows or business processes. It enables the creation and deployment of web applications that can manipulate and synchronise data for use in other mobile or web applications.

As a result, companies with a fully developed digital digital offering can offer their customers access their services and brands via multi-channel multi-channel digital interfaces that can be adapted to all available channels, from desktop to tablets and smartphones.

For an optimal user experience, user identification identification should be a one-off process access to all the customer's services during the digital services during their digital journey. To achieve this for authentication: CIAM (Customer Identity & Access Management) which will centralise, manage and monitor user access user access and apply security policies security policies according to the profiles selected.

Users accessing the APIs exposed by the Boomi platform via digital channels will be authenticated and, depending on their authorisation, will be able to consume a greater or lesser amount of data.

Depending on the constraints in terms of data performance and availability, it is advisable to set up a cache to serve real-time APIs efficiently, as the quality control processes are often complex and cumbersome. In this way, the creation or update via the dedicated API will be sent to the platform and stored in the Real Time cache. This data will be immediately available for the rest of the customer journey. NoSQL solutions such as Mongo- DB, Couchbase or Cassandra. Ultimately, the digital journey is freed from the impact of the complexity business rules or processes.



Stage 2 :
Data activation within the IT ecosystem

2.4. ORDER DATA

The 3rd way of activating data is to give it meaning giving it meaning by ordering it according to analysis or reporting structures, or simply to create specific views.

These are the 2 most common uses, which complement each other:

- Analytics, which enables data to be aggregated in the form of reports and explore in different ways.
- Data Science, which makes it possible to analyse interpret large volumes of data and reveal trends, correlations and patterns beyond pre-built algorithms.

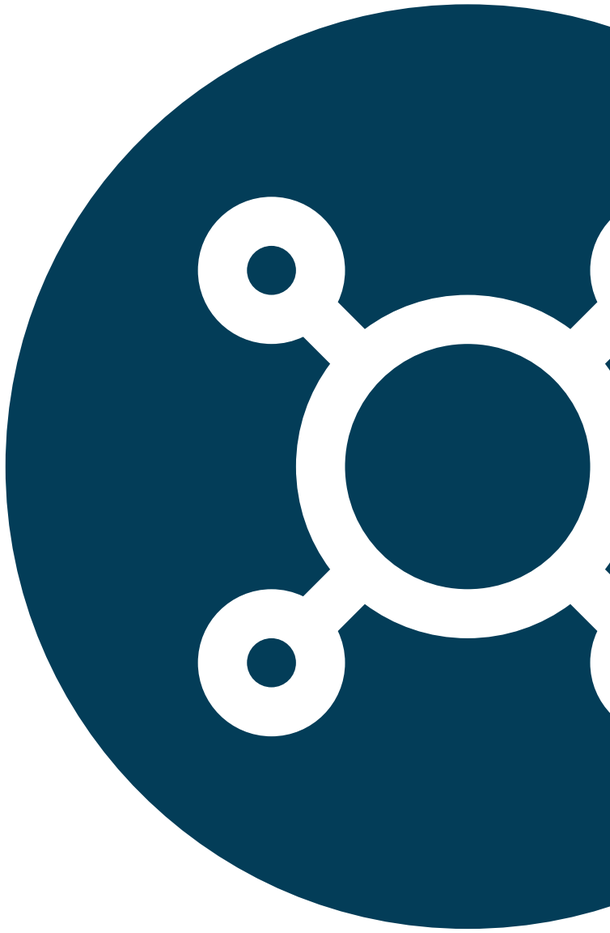
Obviously, this activation is strongly correlated with the choice of underlying architectures, such as relational data warehouses, a NoSQL database or an HDFS data lake. This is why the platform can be plural in terms of storage.

But increasingly, exploration and analysis activities are based on tools that unify access by covering a wide range of connectivity and query languages. PowerBI for Analytics and Databricks for data science are common solutions that offer this performance.

However, it may be necessary to add an transformation step (without recopying or or datamart) in order to present the platform's data from the platform in formats that allow certain types of analysis or visualisation.

To achieve this, the above-mentioned tools will be based on functions that enable each user to create a specific image of shared data, or even cross-reference it with their own data:

- Data preparation, which makes it possible to transform data on the fly, without recopying or datamarting (PowerBI, for example, via its DAX library).
- Data virtualisation, which enables federated governance principles to be applied to heterogeneous data. The exposed model is composed of different layers, which allows the reuse of common entities. These common entities are carried by the map layer of the "Data Centric" model, while the specific layers meet the specific needs of the various business lines. This allows designers to easily reuse reuse data from other domains, thereby reducing the integration workload.



Stage 3 :

Adressing use-cases in dedicated modules

Each business has its own operational needs in terms of data consumption.
Let's take some examples of use cases from the domain customer :

- Propose new selection criteria to increase the audience by cross-referencing campaign history with customer profile, values, etc.
- Manage sales pressure by eliminating certain customers who have already been approached rules.
- Detect fraudulent behaviour and alert / block services to suspicious customers to prevent malicious acts. malicious acts.
- Anticipate dissatisfaction by monitoring networks in predictive models and alert to increase responsiveness in managing conflict management and resolution.

There are 2 options for addressing these needs operational :

3.1. PROVIDE A BASIS FOR DEDICATED APPLICATIONS

It is not always necessary or appropriate to to reproduce and synchronise data from the platform in third-party applications. On the on the contrary, it is simpler for these applications use the platform as their own database.

This may be the case for a CDP (Customer Data Platform), for example. Its many functions: segmentation, targeting, animation, etc. call up the platform's data via webservices and/or connectors. In this way, business users, such as marketers for the CDP, can use all the company's data as levers for action, whereas traditional applications are reduced to remodelled attributes in their own database.

NB: It should be pointed out that the centralised platform is not the only supporting database. This is where business-specific databases come into play, in addition to the enterprise platform.

Stage 3 :

Adressing use-cases in dedicated modules

3.2. SPECIFY DEDICATED MODULES

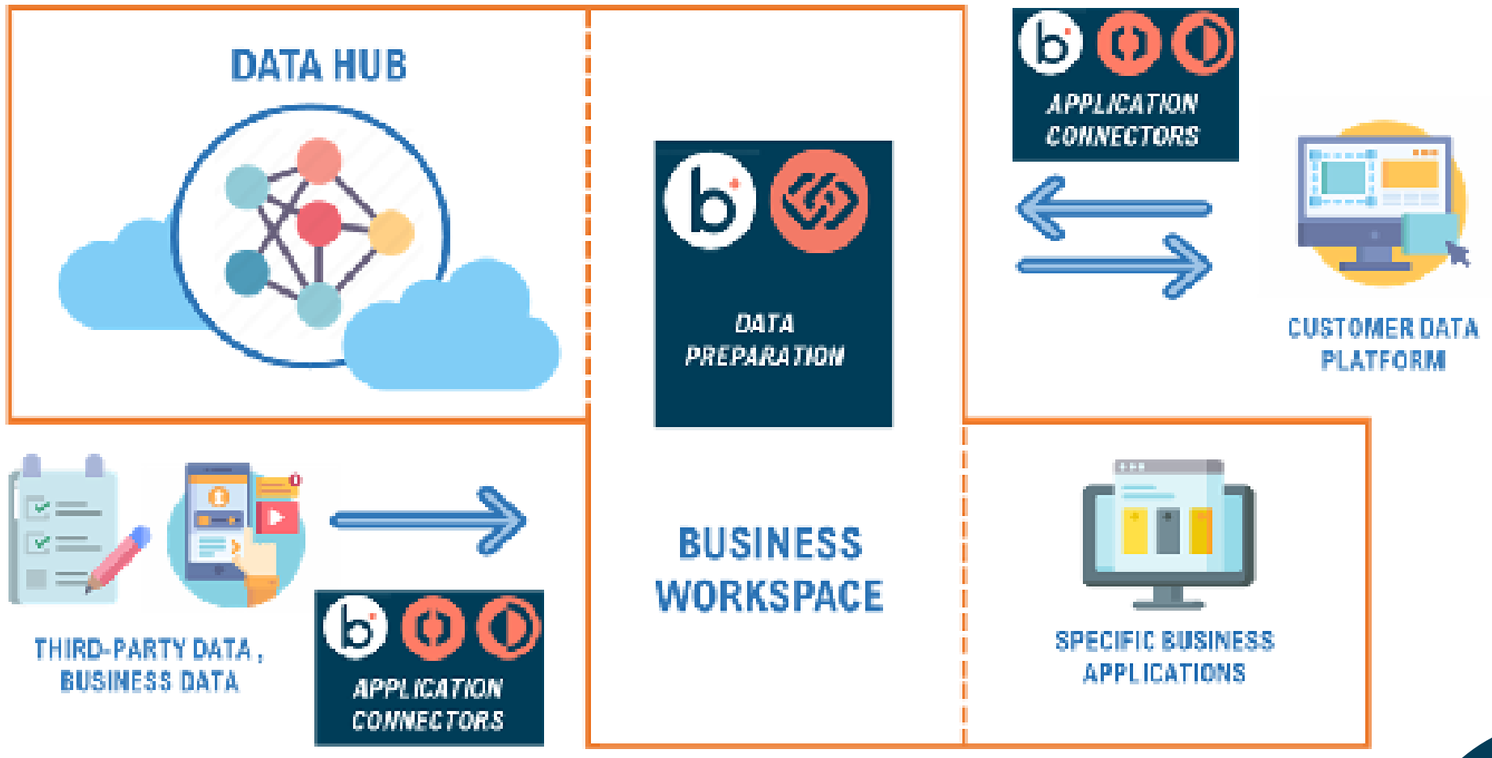
When no business application covers these needs, it's a good idea to create business intelligence and business-specific functions from customised data layers that are like extensions to the centralised platform.

In this case, it makes sense to create specific business environments, based on dedicated visions, in which metadata specific to the and populate the metadata specific to the use cases. Boomi module makes it possible to define specific objects raw data from the centralised platform platform or from local sources. In practice simple click, you can :

- Join several data sets.
- Apply filters to unwanted values.
- Add calculated attributes.
- Aggregate information and build their own indicators.
- Plan and monitor file transformation/export tasks.

Based on the preparation of these data, we can all kinds of processes and functions, including functions, complex use cases that incorporate user interfaces, etc.

Example of a Data platform and activation schemes.



Stage 4 :

4. Facilitating the deployment of the platform : « technical » use-cases

In addition to business applications, the Boomi platform accelerates the implementation of the data-centric platform itself. This is illustrated by illustrates this in a use case common to all Data projects.

We have shown how it is preferable to give priority to exchanges as they occur. Let's take a look at how Boomi can be used to handle the initial loading of data in this way:

The cycle of a Data Centric architecture involves loading data as and when new data-providing systems are added.

The initial loading of data from a new system is very often necessary to maintain a consistent data perimeter, without which the exploitation of data downstream of the architecture (e.g. BI, campaign management) would become ineffective. The main difficulty with the initial loading comes from the volumes involved in these exchanges and the time needed to extract, transform and then persist the data, which has the effect of degrading service levels or even interrupting them. Information systems often use ETL tools to solve these high-volume problems of high-volume processing in batch mode. This approach has the disadvantage of duplicating the integration effort required, because the batch vs real-time dichotomy often means re-implementing the same logic as for message-oriented tools, not to mention the potential licence costs.

It is therefore logical to look for a solution that address the 2 use cases: initial loading and ingestion of changes over time. over time. The AtomSphere platform offers several advantages in this respect, as it enables us to deploy workflow processing on a set of atoms called molecules, which, in conjunction with multithreaded execution , enables performance in excess of 10,000 transactions/minute. Once the initial load has been completed, the architecture can quickly be returned to its nominal configuration. In this way, the extra cost of provisioning the additional hardware resources used during the initial load is only temporary.