



## **ARCHITECTURE OVERVIEW**

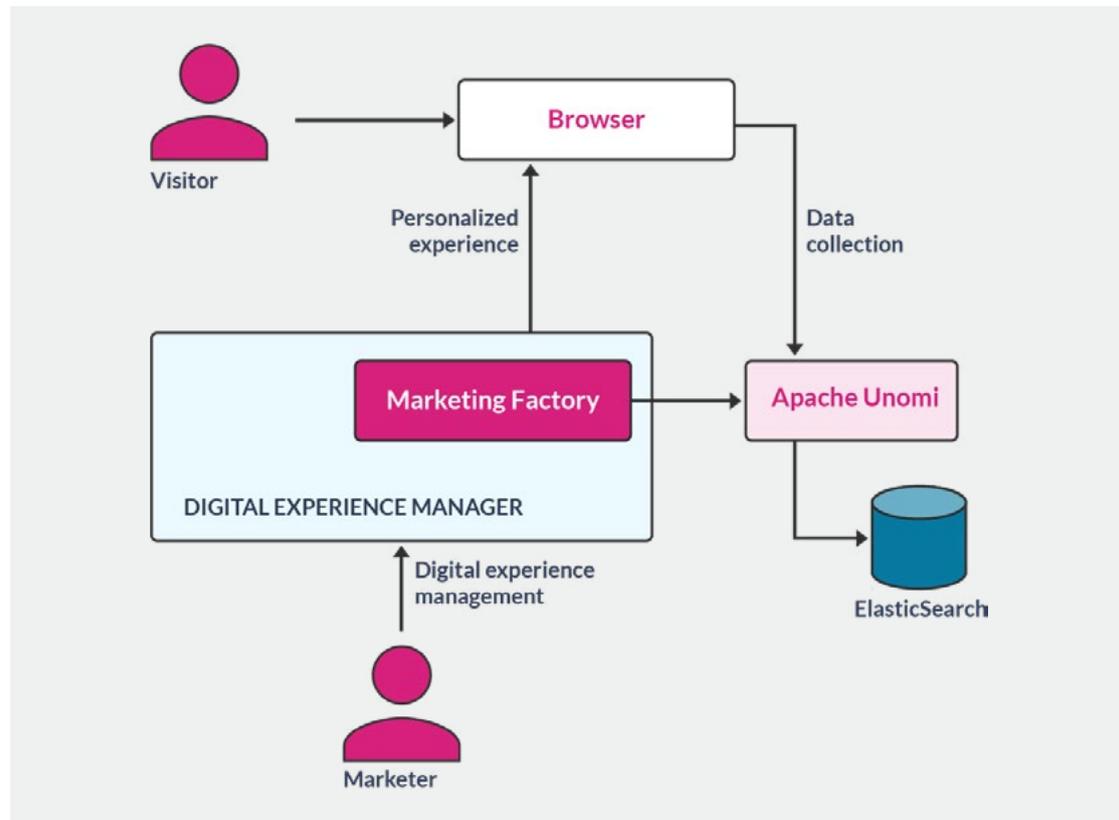
The personalization of the customer experience is in every marketer's mind and this requirement has strong impacts on customer data integration, across channels and applications. **This document will help developers and architects understand how they can use and extend Marketing Factory and Apache Unomi to integrate customer data.**

### CONTENTS

- › Architecture Overview
- › Marketing Factory in details
- › Context server specification
- › Apache Unomi in details
- › Extending Marketing Factory and Apache Unomi

# High level architecture overview

## Concepts



### Digital Experience Manager (DX)

A customer digital experience platform that contains CMS, WCM, Portal features as well as a built-in developer environment called the Jahia Studio. It is extensible by adding or developing OSGi modules that may range from simple content components to full-blown Single Page Applications

### Marketing Factory

Marketing Factory leverages the Jahia DX platform and Apache Unomi to provide an AB Testing and personalization solution.

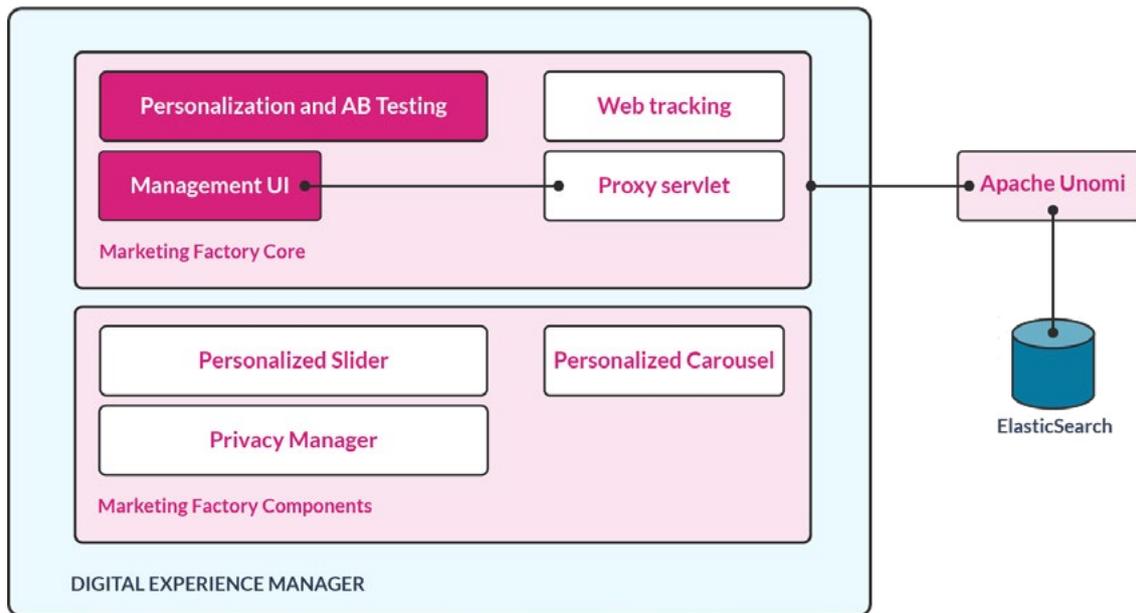
### Apache Unomi

The core Customer Data Platform engine. Apache Unomi is where all the customer data as well as all the events generated by visitors are stored and indexed.

### ElasticSearch

The backend storage, indexation and query server system to store all the data in Apache Unomi. It is also highly scalable and offers powerful high availability mechanisms.

## Marketing Factory In Details



### Marketing Factory Core

Built as a Jahia DX module, the core of Marketing Factory interfaces with both DX and Apache Unomi to provide its functionality. It is composed of:

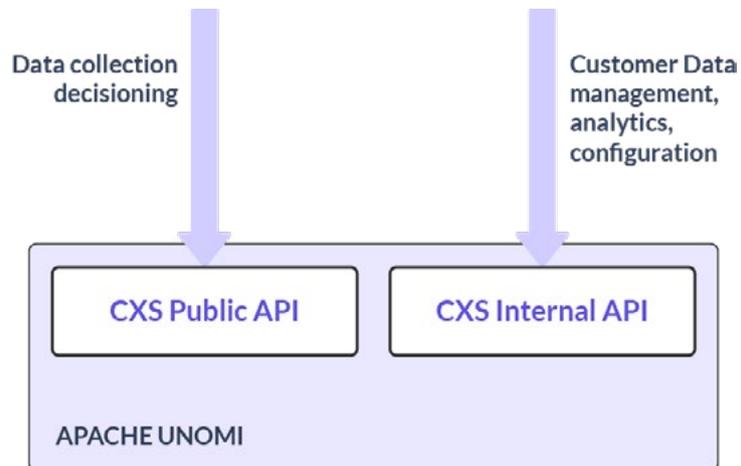
- › Web tracking: Marketing Factory automatically inserts in pages generated by DX the necessary web tracking JavaScript code that will track visitors and collect events and send them back to Apache Unomi.
- › Personalization modules: these are strongly integrated with DX's edit mode to offer personalization as well as A/B testing features on any DX component.
- › Management UI: The interface for marketers to access visitors data, profiling, analytics and data integration features
- › Proxy Servlet: This proxy servlet is used to make sure that only Jahia DX users with the proper credentials have access to the CXS Private Admin REST API implemented by Apache Unomi. It serves as an authorization bridge between the two systems.

### Marketing Factory Components

These components are elements designed for Jahia DX that will integrate with the personalization and A/B testing features of Marketing Factory to build experiences that contain personalized banners, a carousel or a privacy manager. New components may of course be developed by third parties.

## A standardized interface - The Oasis Context Server Specification (CXS)

The interface between Marketing Factory and Apache Unomi is defined by a standard called The OASIS Context Server specification. It is a work in progress at the OASIS foundation, defining an open standard to enable the delivery of personalized user experiences. The goal is to assist organizations that currently struggle to create and deliver consistent personalized experiences across channels, markets and systems. The specification defines a standardized API contract to talk to context server implementations.





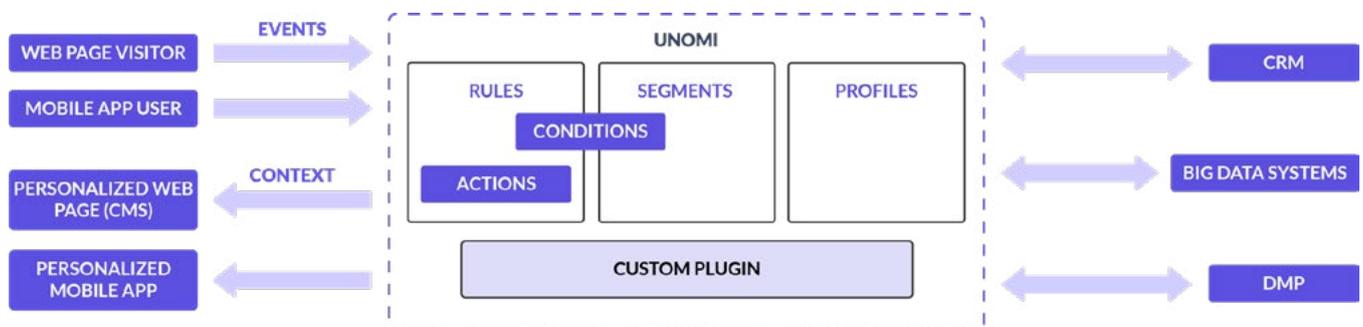
## The open source customer data platform

### What is Apache Unomi ?

Apache Unomi is the [Open Source Customer Data Platform](#) and an implementation of the OASIS Context Server specification. It processes customer-generated events, executes rules based on the content of events to update customer profiles. It takes care of storing and delivering the profile in a context object that also contains augmented information such as the profile's resolved segments, or collected properties aggregated from different systems connected to it (CRM, WCM, etc..).

Apache Unomi is an API-first server and is used by Marketing Factory to track customers, deliver AB testing and personalization but it could also be used to deliver ads, or even to build personalized native mobile applications.

### Functional architecture



#### Events

The main input into the Apache Unomi server are events. Events may range from a single page view event to a native mobile application actions such as a button pressed or an application being started. New applications may define new events types and simply send them through the CXS Public REST API using HTTP POST calls to have them processed and recorded by Apache Unomi.

#### Conditions

Conditions are used to match against events or profiles and may be used to build segments or rules. An example of a condition may be: profiles that have an age property that is above 21 or a profile that has seen a page or a group of pages. Conditions may also be combined using AND and OR operators to build more complex conditions.

#### Context

A context is a collection of the current customer profile as well as other contextual information about the current interaction with Apache Unomi such as the customer's segments, session information (i.e. resolved geographic location from the IP), etc..

#### Segments (Dynamic)

Segments are used to segment the profiles stored inside Apache Unomi. A segment is defined by defining conditions that a profile must match for it to be included in the segment. As the profile may change (or the events it is associated with), the segments are dynamic groups that may see profiles come and go depending on the conditions defined.

#### Profiles

Represent collected information about a customer. By default a cookie is used to keep track of the profile on the browser side. Profiles may contain extensible properties that may contain personal or anonymous data. By default no personal data is required for the tracking mechanism to work, a profile ID is automatically generated when a new visitor is detected by Apache Unomi.

## Static Lists

Profiles may also be either manually or adding to lists using rule actions. These lists are not dynamic, meaning that once a profile is added to a list it will only be removed by manually editing the list to remove the profile from it.

## Rules

Rules are composed of conditions and actions. If a rule's conditions are satisfied, the list of actions defined in the rule will be executed. Rules may be defined and modified at any time, and will react to events being inserted into the system. As conditions and actions may be extended using plugins, the rule engine makes it a very powerful way of adding realtime functionality to Apache Unomi.

So, in order, the behavior of the rule engine is the following :

1. An event is received
2. The rule engine is triggered by the reception of any event (internal or external)
3. For each rule (in order of priority):
  - › evaluate its conditions
  - › if they match, the actions are executed in the selected order they are defined

## Scoring Plans

Scoring plans are a digital marketing tool that allows to define conditions that, if met, will increment a score on a profile. Again the condition system is used here, and complex conditions may be used to increment scores on profiles.

## Custom plugin

A custom plugin may be implemented by developers to provide new conditions or actions, as well as deploy predefined rules, property types, or any other functionality that may be implemented using OSGi bundles (Apache Unomi is itself an OSGi application).

## Actions

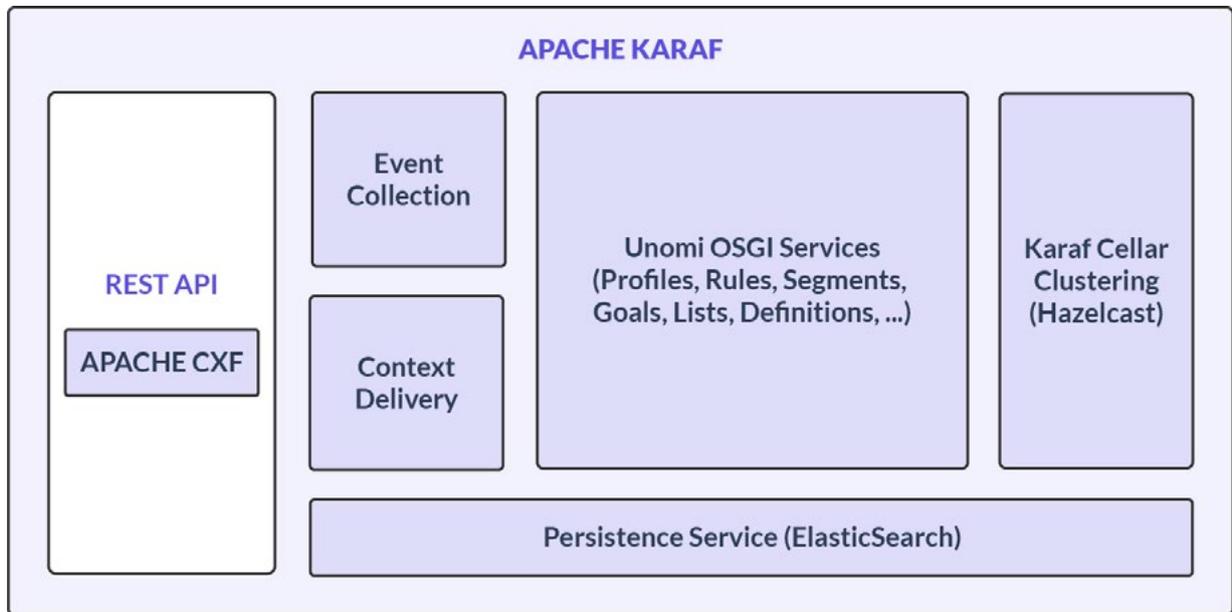
Actions are pieces of logic that may be executed when a certain set of conditions is met. Actions are always the second part of a rule, that will execute only if the conditions of a rule are satisfied.

### Built-in actions:

- › Update profile properties
- › Add to list
- › Send an email
- › Push or pull data to Salesforce CRM
- › Enrich customer session with weather data

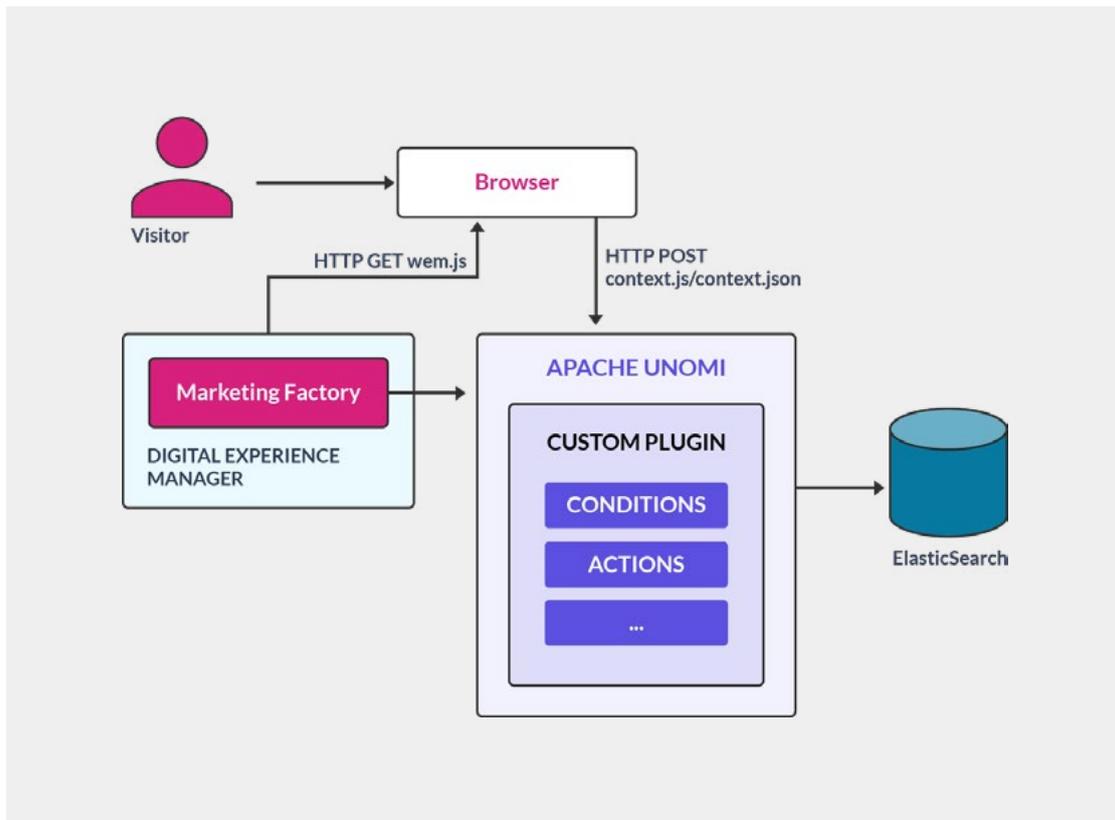
## Technical architecture

Apache Unomi is a Apache Karaf (OSGi) application. It runs as a server-based Java service that is accessible through a REST API that is being defined as part of the OASIS Context Server specification work. As it uses an OSGi engine at its core, this means that it is designed to be extensible and very flexible in terms of deployment scenarios. It is also designed to scale out since it uses cluster-tested technologies such as ElasticSearch, Apache Karaf Cellar (based on Hazelcast) and REST-API level load distribution.



The REST API is built using Apache CXF based on Java Beans that are automatically mapped as REST endpoints. Custom plugins may of course define new REST APIs to extend the built-in API functionality.

## Extending Marketing Factory and Apache Unomi



### Client side

#### Javascript

**Apache Unomi context.js/context.json:** This endpoint may be used to retrieve the visitor's profile, as well as directly inject events and also evaluate conditions to check if the current visitor matches some personalization conditions (for example: only display this text if he belongs to segment XYZ)

**Marketing Factory wem.js:** Use wem.js functions to send data to Apache Unomi + catch events generated when a variant is displayed (and send to your analytics platform). This event is notably used by our AT-Internet and Google Analytics modules.

#### Cookies

**context-profile-id:** this is the identifier of the cookie that is used to keep track of the Context Server/Apache Unomi profile. This identifier should be conserved as much as possible, otherwise tracking will not work properly. It is a value that rarely changes, only in the case of profile merges it will be assigned a new value of the resolved "master" profile.

**wem-session-id:** this cookie is used by Marketing Factory to keep track of visitor sessions. It should be short-lived and represent an actual group of interactions that will be grouped together for session-based analysis in the various management UIs.

## Server-Side

### Apache Unomi plugins

A custom plugin may be implemented by developers to provide new conditions or actions, as well as deploy predefined rules, property types, or any other functionality that may be implemented using OSGi bundles (Apache Unomi is itself an OSGi application).

Such plugins can send data to other third-party systems such as Big Data technologies, CRM, DMP, Marketing automation, or any system that may be interested in processing customer generated events and profile information.

### Conditions

New conditions may be defined by custom plugins, and may for example be used to integrate with external systems to provide more ways to make with visitors. Conditions may either be implemented in Java as JSON definition files that extend existing conditions with more specific conditions. Conditions may be used in segment definitions or rules.

### Rule actions

Actions are implemented in Java, and new actions (that may push or pull data from external systems) may be implemented as part of a custom Unomi plugin. For example, our Salesforce CRM integration is built using actions that may pull or push lead data from and to Apache Unomi profiles.

### Other plugin contents

Plugins may contain a lot more, including property type definitions, predefined rules, custom REST endpoints or anything that is acceptable in an OSGi bundle (activators, listeners, services).

## About Jahia

Jahia is a software vendor that provides the most integrated open-source Java Digital Experience Platform (DXP) that unifies content, applications and customer data for truly personalized customer experience.

Leveraging Jahia's DXP strong continuous integration capabilities and easy to use interfaces, IT & Marketing teams build and manage together all their multichannel initiatives, faster than with any other solution.

Founded in 2002 and headquartered in Geneva, Switzerland, Jahia Solutions Group has its North American headquarters in Washington, DC, with offices in Houston, Toronto and throughout Europe. Jahia's loyal customers community counts hundreds of global brands and organisations such as HomeAway, BNP Paribas, Sodexo and the European Parliament.

## Contact us

Group HQ – Geneva  
Switzerland  
T +41 22 361 34 24

North American HQ –  
Washington, D.C.  
T +1 202 656 7874

France – Paris  
T +33 1 44 79 33 79

Austria – Klagenfurt  
T +43 463 287 008

Germany – Freiburg  
T +49 211 178 377 36

Canada – Toronto  
T +1 905 257 7471

[info@jahia.com](mailto:info@jahia.com)

Contact form 

Demo request 

## Resources

[www.jahia.com](http://www.jahia.com)

[academy.jahia.com](http://academy.jahia.com)

[store.jahia.com](http://store.jahia.com)

[youtube.com/JahiaCMS](http://youtube.com/JahiaCMS)