WalkMe Performance Best Practices

Brief Overview

WalkMe loads all its resources asynchronously meaning that it doesn’t hold the site in order to load. Since WalkMe optimizes performance in this way, it won’t have a visible impact on the end user experience or the site’s performance.

The WalkMe Player performance is divided into two parts:

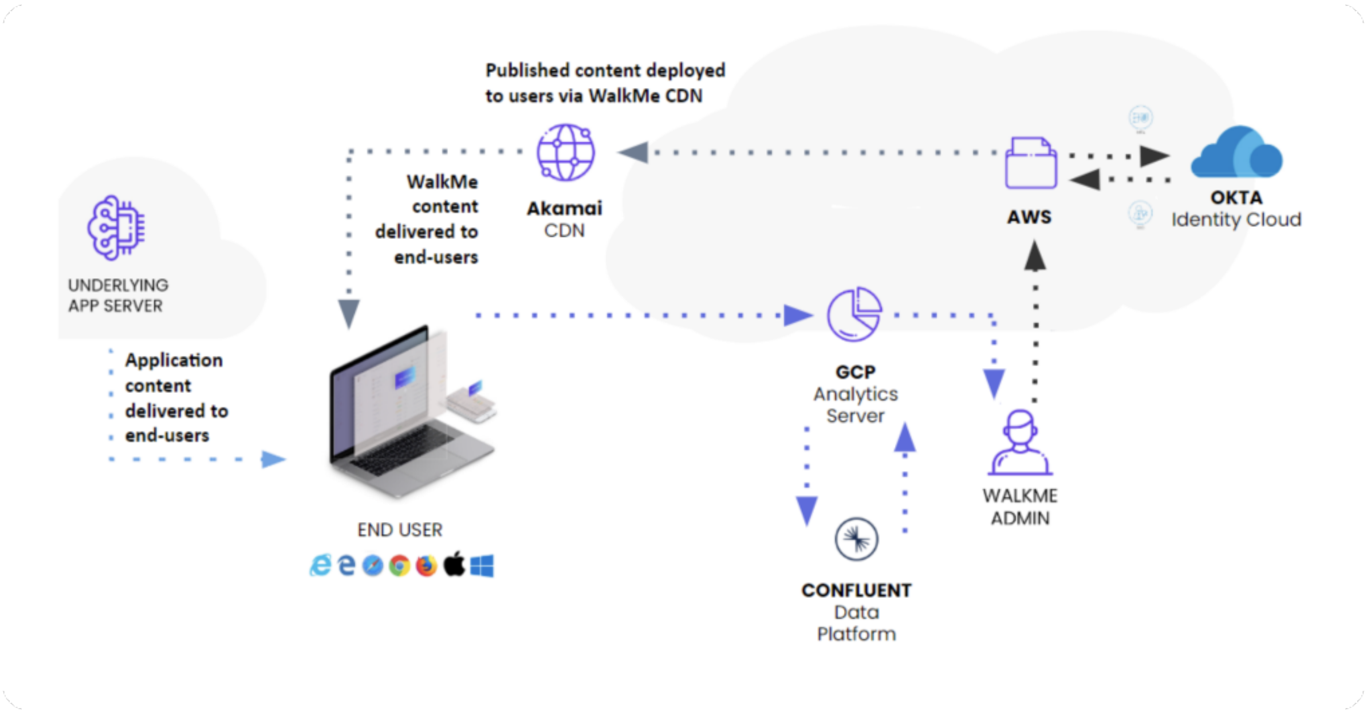
1. Loading performance
2. Runtime performance

Loading Performance

WalkMe runs on the browser and uses browser resources in order to load. The speed in which WalkMe loads depends on the network connection of the user. A faster network connection will make WalkMe load faster overall.

WalkMe also splits its files into multiple smaller ones to avoid large files from needing to load all at once.

The WalkMe engine is based on static resources that are cached on the browser and the WalkMe Content Delivery Network (CDN). The CDN delivers the static resources that are closest to the end-user’s physical location.



Published content deployed to users via WalkMe CDN
WalkMe content delivered to end-users
Akamai CDN
AWS
OKTA Identity Cloud
UNDERLYING APP SERVER
Application content delivered to end-users
END USER
GCP Analytics Server
WALKME ADMIN
CONFLUENT Data Platform

Runtime Performance

WalkMe by itself or with published content naturally loads very efficiently. To ensure WalkMe continues to perform and load in the best possible way, it's important to build WalkMe content with optimizing performance in mind.

Page load evaluation

By default, WalkMe checks segmentation and engagement/auto play rules upon page load. If the site is a single page application, WalkMe checks these rules based on the configuration you set in the SPA (Single Page Application) tab of the Editor. This check is called page load evaluation.

This is important because the number of page load evaluations and the way they are configured can impact the speed at which WalkMe loads upon page load.

What WalkMe checks on page load:

- **Segmentation tags:** Segmentation allows you to decide when and where to display your WalkMe content. Read more.
- **Auto play rules** for ShoutOuts, Smart Walk-Thrus and Surveys. Read more.
- **Display rules:** Logic statements inside of the Rule Engine based on dynamic or static attributes. Read more.
- **Widget segmentation:** Segment where and when the WalkMe Widget displays based on options available in Rule Engine. Read more.

- **UUID:** Detect any unique user identifier, if configured. Read more.

Continuous evaluation

WalkMe checks additional items like goals, element behavior, or segment evaluation even if you haven't navigated to a new page. This is called continuous evaluation.

The number of continuous evaluations and the way they are configured are important to keep in mind from a performance standpoint.

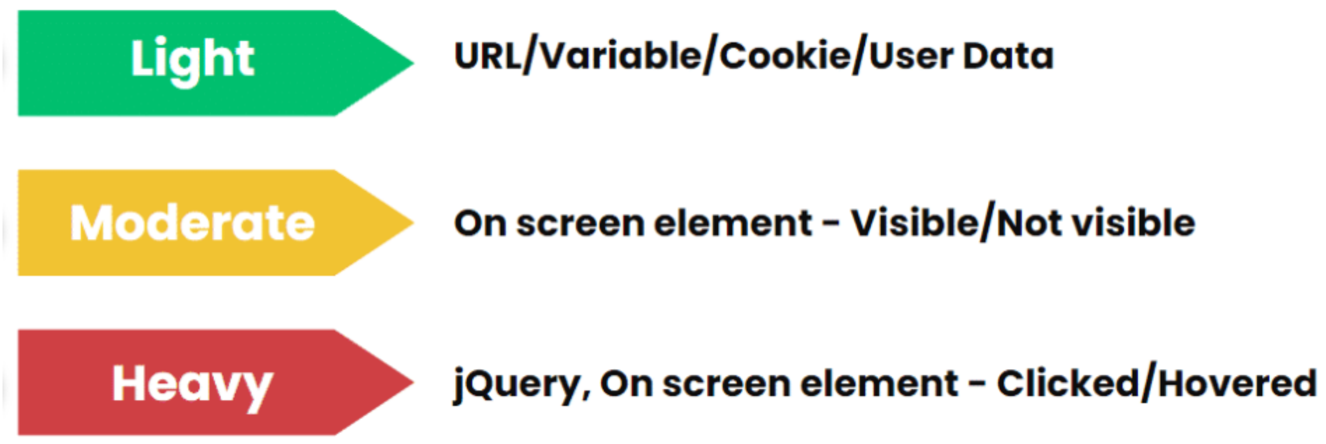
What WalkMe continuously evaluates:

- Selected element options, if the radio button is selected:
  - Appears after page load and then stays visible
  - Appears and disappears as a result of user action
- Display conditions, **if the "constantly check" is toggled on**
- Smart Walk-Thru and Walk-Thru Goals (for 2 hours after starting the flow)
- Onboarding Goals
  - Tracked all the time regardless of whether the Smart Walk-Thru was initiated

Build to Optimize WalkMe Performance

Rule Engine

The Rule Engine can be used when building for things like auto play rules, goals, and segmentation. Consider the following hierarchy when creating rules to optimize the performance of your conditions:


Light
URL/Variable/Cookie/User Data
Moderate
On screen element - Visible/Not visible
Heavy
jQuery, On screen element - Clicked/Hovered

Rule Engine Condition Performance Hierarchy

In general, the fastest conditions for WalkMe to evaluate in the Rule Engine are URL, variable, cookie, and user data. These are rules you want to use first since they can evaluate quickly and display your WalkMe content to who and where you want it.

Slower conditions to evaluate include on screen elements and jQuery.

Using slower conditions doesn't necessarily mean they are going to be visibly slow to evaluate. But, knowing that statistically some rules are easier to check quickly, it is best practice to ensure that a fast condition is always present. If a fast condition is not an option, try to limit the number of rules you have using only a slower condition.

Performance rules to live by when building

1. Always try to prefer light rules over heavy rules (assuming both are applicable)
2. If you have a heavy rule, always accompany it with a light rule – that way if the light rule is false WalkMe won't even try to evaluate the heavy rule
3. Try not to include too many heavy rules

Segmentation

WalkMe will only initiate the search for on-screen elements and load an item when the segmentation is true. Segment tags attach an invariable rule to a group of items. For example, a user group or office location. Segment tags are very efficient because they allow WalkMe to evaluate a rule only once for a large group of items. If you have a number of items that follow the same segmentation logic, use a segmentation tag rather than reusing the same display conditions on multiple items.

Continuous evaluation for SmartTips and Launchers

If the condition set in the Display Condition is False on page load, the Launcher / SmartTip will not be drawn on the screen. Use the "Appears and Disappears" option (read below) to tell WalkMe to try again. Every time WalkMe checks whether to play the SmartTip or Launcher – it first assesses the Condition.

If on page load the condition set in the Display Condition is True, but might later change to false (without page refresh) – use the "Constantly check" option to tell WalkMe to remove the SmartTip/Launcher when it's False.

Use this function wisely and only when absolutely needed; adding this to more than a handful of SmartTips could lead to slower loading times. If you need to add this functionality to a lot of SmartTips, QA the end user experience to check for any possible lag.

Number SmartTips and Launchers

There is no specific limit to how many SmartTips or Launchers you can have on one page because this varies between different sites. Create as many SmartTips and Launchers as your site requires, with the understanding that the more evaluations WalkMe needs to run on a page, the longer those evaluations will take, which may result in WalkMe content taking time to appear on the page.

Impact on the Editor

There is no limitation on the number of SmartTips that can be in a SmartTip set. However, if there are too many, it may lead to performance issues and cause the Editor to crash. It’s difficult to provide a specific number as it depends on various settings and conditions.

Please use your best judgement and consider the potential impact on the Editor functioning if too many SmartTips are added to a set.

To ensure an optimized experience, follow these guidelines:

- All SmartTips within the same set should display on the same page. The SmartTip set display rule is evaluated only once upon page load and the individual SmartTip display conditions are evaluated according to its specific configuration and selected element.
- Optimizing the actual rules of the SmartTips and Launchers is more important than the number of SmartTips and Launchers themselves