

# Jproxy Cache

This module provides you with a **proxy system** that can **process** all the **traffic** that comes into or goes out of your **Booking Engine** through the **Webservice or the Call Center** modules. It includes a **cache system** with sophisticated rules criteria that will **determine** in each case whether to **store the generated response or not**, and thus, take advantage of those requests that are repeated.

## Who could benefit from this module?

**Hotel Chains, Wholesalers, DMCs, Bed banks**, and any **company** that is **providing accommodations** through the Webservice Booking Engine.

## Why contract this module?

In the current environment, your system needs to be prepared for unexpected punctual increases of traffic, and generally, it should **answer requests in optimal response times** for an important percentage of cases. This way your clients will have your **product details before your competitors** can supply their response. In this way, your company will be more competitive.

With this module, your **system** will be more **efficient** and **effective**. As it is integrated with Juniper's **Booking Engine**, it is the **best choice** for integrate or not a cache system with your Booking Engine.



**90%**

Responses stored in cache



**+80%**

Coincidence in searches



**<10ms**

Cache search processing



**0ms**

Added to responses



**50.000**

Concurrent transactions

Contact us to request a demo



SCAN ME



# JProxy Cache

Optimize the performance of your system



# Juniper JProxy Cache

- Drastically **reduce** your **response time** when the results are in cache.
- **Keeps under control any excess traffic**, thus keeping your **system healthy**.
- **Fully integrated** in your **Juniper Booking Engine** solution.
- Dynamic **rules** that **optimize** the **cache** usage.
- **Intelligent self-learner system** that **adapts** itself to the search **trends**.
- Works in a **transparent way without adding latency** to the search responses.
- **Improved look-to-book** ratios with your suppliers.
- **By us optimizing the Cache timeouts**, you minimize price changes during the booking process.

## REQUESTS

- Server **responses go directly to the client** and they are **processed** in parallel, **without affecting response time**.
- The **search time** in Cache is **negligible**, becoming a transparent process.
- If the **response** is indeed in cache, it is **returned immediately**, otherwise, the normal process is followed.

## RESPONSES

- Server **responses are returned to the client** and are processed in parallel, **without affecting the response time**.
- Based on the **dynamic intelligence rules**, it will be **decided** whether the result should be **stored in cache, or not**.
- It is a **dynamic and intelligent process**, capable of learning from prior search behaviour, and thus, **adapting** the rules to the **latest trends**.

