

**Technology**

**V1.1**

**4/11/2015**

**Overview**

BookingPal core architecture is built on the Razor Platform. BookingPal purchased the IP for the Razor Platform in 2013. The original core platform from Razor covered the following key areas:

* PMS Portal
* Agent Portal
* PMS API
* Availability Calendar
* Pricing Table

Since the purchase of the platform BookingPal developers have added or enhanced the following areas:

* Online Booking Processing
* Payment Gateway integrations
* Channel Partner API
* Advances PMS integrations
* Updated Availability Calendar
* Updated Pricing Table
* PM Portal
* Admin Porta
* Channel Portal
* Registration Process
* Booking and Calendar Widgets

The architecture is built using Linux, Java and PHP, MYSQL serves as the primary DB. Eclipse is used as our primary IDE environment. We use Jira as our Bug tracking and task management system. We practice the Agile development method and specifically the XP Agile method for development.

The platform is separated into 3 main functions:

* Core Platform
* PMS Interface
* Channel Interface

*Core Platform*

The core platform handles all the transaction processing, this consists of:

* Availability Calendar
* Pricing Engine
* Yield Rule Engine
* Booking Engine

The core platform is maintained by our team of developers in the Ukraine with additional support from US and Serbia. All coordination for core updates are done through the lead developer in the Ukraine. All new features and functions that are required for this platform are developed by this team.

*PMS Interface*

This interface supports all PMS partners. When Razor was purchased a PMS API interface was available as part of the base platform. However, as we started to do PMS integrations we realized that this PMS API interface could not support the challenges for each PMS. Currently every PMS integration is a custom integration and so we have divided this interface out from the core platform.

PMS developers have a base set of functions to use and they modify or manipulate the data to fit into these basic set of functions. If new requirements for PMS providers are required this is discussed and architecture is reviewed and requirements are delivered to the Core development team. We will be developing a new PMS API this year to cover the features we have added to platform. This is a lower priority for us at this time since most of the PMS providers do not have the technical skill sets to integrate to an API.

The PMS integration team is managed by a senior developer in the Irvine office. The integration team consists of engineers in India, Serbia and the US.

*Channel Interface*

The channel integration process is primarily done through our integration API and our library of widgets. The API is constantly updated as needs arise and new features are added to the core platform. All new functions or changes to the API are backward compatible so no existing user is impacted by any change.

We currently do two types of integrations depending on the type of channel partner. Some of the larger channel partners require us to integrate directly to their interface. In this case we have our development engineers work with the channel partners API’s. With some of the smaller to mid-tier channel partners we provide our integration API and our building widgets for them to build out their sites. For these channel partners we also have an integration engineer assigned to them to work through the integration details.

All of our channel integrations are currently done through our development engineers in our Irvine office.

**Architecture**

We host our platform at Amazon cloud service in the US. We currently have two regions hosting the platform the East and the West Regions.



The East region is our primary data center and data is replicated to the west region. The two regions are identical with the exception that the Admin portals are connected to our primary DB located in the east region. All searches are performed at the local regionally level. All PMS property data is loaded to our primary DB in the east region and replicated to the wet region for local searches.

Each regions uses an HAPROXY to perform load balancing and more servers can be added as load demands.

Currently all of our transactions are written directly to our primary DB. Which is then replicated to all of our other DB’s. To improve on the transaction processing we will be moving to an ASYNC transaction processing method this year. This will remove our channel partners on waiting for the DB and our PMS providers from being in the transaction process.

Currently all of our searches are being performed out of one of our standby DB’s. Eventually we will be moving the availability calendar and pricing tables to an in memory cache, either on each of the web servers or to a cache farm.



Each WEB server is connected to our PMS channel and to all of our available payment gateways. These connections provide for real time quotes for properties and for online booking of properties. As our load increases we can add more WEB servers and more connections to our PMS partners and our payment gateways.

**Searches**



We provide two methods for channel partners to perform searches on our platform either through our API or through the consumption of our XML file. For searches we categorize channel partners into three primary buckets large, medium or small demand. By categorizing the partners we are capable of building out different interfaces for each. For large and medium size channel partners we push the search data to them and allow them to handle the search load on their local infrastructure. By doing this we can keep our infrastructure at a relatively small footprint.

For larger partners such as Booking.Com, Expedia, AirB&B… they require us to load properties into their systems and searches are done on their servers. For these types of large channel partners they perform the searches locally on their servers and we update them periodically with our pricing and availability calendars. The upload of data to these partners do not put a burden on our servers and we can dedicate servers to each channel partner if required.

For the medium size channel partners that do not have an upload interface we provide an XML file that can be consumed from our SFTP site. The XML file provides the availability calendars and pricing structures required to perform searches. These channel partners typically have significant number of searches but are below what a top tier channel provider is capable of performing. These channel partners also require very minimal load on our servers and we can also provide a dedicate server to each of these channel partners if required.

For the smaller channel partners that do not have local cache capabilities on their servers we provide an API or an iFrame solution. Currently this load is very minimal on our servers and if we had to we could scale out to more web servers.

If a channel partner is capable of supporting delta changes to the availability calendar and pricing table we can then also push these changes real time. This is done so a complete load of all properties is not required for the channel partner. We can support this method even if the PMS does not support real time updates. Depending on the PMS provider we typically get property updates once a day. So this forces us to push all property updates to our channel partners only once we have received the updates from our PMS providers. In some cases we can push delta changes to the channels in our cases we have to push the complete update.

Since we push most of the search requirements to our channel partners to perform locally on their platforms we do not have a large load on our platform. However, we believe that this could grow and we do not want to add more replicated DB’s. Our long term vision is to provide an in memory cache solution for our availability calendars and pricing.

**Transactions**



To complete a transaction on BookingPal two checks must be performed.

1. Check to see of property is still available and retrieve latest price
2. Book the property

Since most PMS providers do not have the ability to give us live calendar and pricing updates we must check with them prior to booking a property. We must also confirm the booking with them once a booking request is made.

Our long term thinking is to move the Booking process to an ASYNC process so the channel is not waiting on the confirmation of the booking. We will provide a pending booking confirmation to the channel while we wait for a confirmation from the PMS provider. Most channel partners are not capable of handling a pending booking transaction at this time. So we must insure that the booking is confirmed by the PMS provider before the channel confirms to the renter.

**Payments**

There are four methods for payments to be collected during the booking process.

1. Channel Partner collects the payments (Channel is Merchant of Record)
2. BookingPal passes payment information to PMS (PM is Merchant of Record)
3. BookingPal Completes payment on behalf of PM (PM is Merchant of Record)
4. BookingPal executes payment (BookingPal is Merchant of Record)

*Method 1*

When the channel partner is the merchant of record all funds are collected by the channel partner and neither BookingPal nor the PM is involved with collection of the funds. For this method the channel partner will collect and distribute all funds to BookingPal and to the PM. The PM also handles all cancellations with the renter. The cancellation rules are typically based on policies that the Channel has setup with the PM and are limited to the rules that are available on their platform.

*Method 2*

Using this method the channel partner passes the payment information to BookingPal through a secure method and the information is passed to the PMS provider. The PM collects all the funds and confirms the booking. BookingPal will invoice the PM for the funds to be distributed to the channel and to BookingPal. We provide email confirmation emails to the renter which contains a cancellation link. When clicked this link will execute a cancellation based on the cancellation rules setup on our system. We will either call the PMS system to cancel the reservation or send an email to the PM to cancel the reservation.

*Method 3*

Using this method the BookngPal will execute the payment using BookingPals connections to the payment gateways. BookingPal will use the PM’s credentials to execute the payments on behalf of the PM. BookingPal will invoice the PM for the funds to be distributed to the channel and to BookingPal. We provide email confirmation emails to the renter which contains a cancellation link. When clicked this link will execute a cancellation based on the cancellation rules setup on our system. We will either call the PMS system to cancel the reservation or send an email to the PM to cancel the reservation.

*Method 4*

Using this method the BookngPal will execute the payment using BookingPals connections to the payment gateways. BookingPal will use BookingPals credentials to execute the payments. BookingPal will transfer funds to the PM and the change once the reservation has been marked completed. We provide email confirmation emails to the renter which contains a cancellation link. When clicked this link will execute a cancellation based on the cancellation rules setup on our system. We will either call the PMS system to cancel the reservation or send an email to the PM to cancel the reservation.

BookingPal does not encourage nor do we push use of Method 4. We do not want to be merchant of record on many of the transactions. We prefer the selection of Methods 1-3 as the primary methods to collect funds. Methods 2-4 require us to meet PCI requirements which we are currently in the process of completing. We are anticipating to complete this process in the next couple of months.