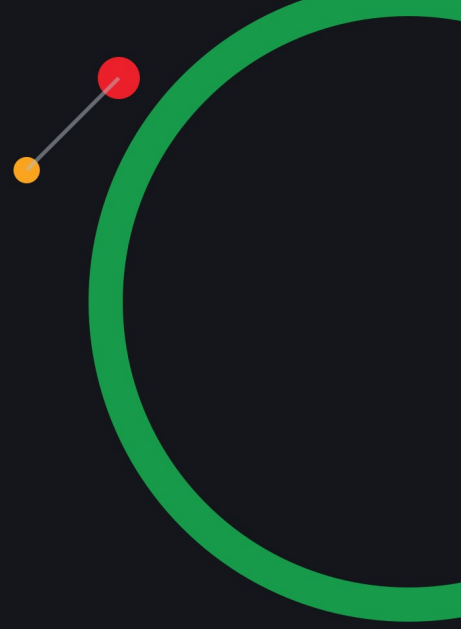




PLAYBOOK

FROM LEGACY TO CLOUD



An OTA Distribution Playbook
for Hospitality Groups

How vacation rental, resort, and campground operators can connect to Expedia and other OTAs — without losing control of their data, rates, or guest experience.

Contents

1. Introduction: Distribution Is No Longer Optional
2. The Hidden Complexity Behind “Just List on Expedia”
3. A Framework for OTA Integration That Scales
4. From APIs to MCP Servers: The Next Shift in Integration
5. Case Study: Scaling OTA Distribution for a Top-3 Travel Booking Platform
6. Common Pitfalls to Avoid
7. Readiness Checklist: Are You Prepared to Integrate?
8. About Keleno & Next Steps

1. Introduction: Distribution Is No Longer Optional

Guests don't start their search on your website anymore. For most vacation rental operators, resort groups, and campground networks, the first touchpoint is an OTA — Expedia, Booking.com, Vrbo — and increasingly, guests never leave that platform to book directly.

That shift has turned OTA distribution from a marketing channel into core infrastructure. The operators winning today aren't necessarily the ones with the best properties — they're the ones whose inventory, rates, and availability show up accurately, in real time, everywhere a guest might be looking.

The problem is that most hospitality businesses didn't build their systems for this. Reservation systems, property management platforms, and channel managers were often adopted at different times, from different vendors, for different reasons. Bolting on a direct OTA connection to that patchwork is where most integration projects run into trouble.

This playbook walks through what that complexity actually looks like, a framework we've used to solve it, a real integration case study, and a checklist to help you assess your own readiness.

2. The Hidden Complexity Behind “Just List on Expedia”

On paper, connecting to an OTA sounds simple: send them your inventory, they send you bookings. In practice, a handful of structural problems show up almost every time:

Data lives in silos

Room counts, occupancy, rate rules, and revenue data are often scattered across separate systems per property or per brand, with no single, current view. Before you can push standardized data to an OTA, you have to be able to assemble it in the first place.

Multiple suppliers, one format

Larger operators pull inventory from several sources — different property management systems, channel managers, or owner-managed listings. Expedia and other OTAs expect one standardized feed, with every mandatory field populated correctly. Reconciling multiple source formats into a single compliant format is a project in itself.

CRS flexibility

Many operators run more than one Central Reservation System, whether from prior acquisitions or regional differences. An integration that only works with a single CRS creates a ceiling on growth the moment you add a new brand or system.

Booking is only half the job

A real OTA connection has to support the full guest journey — search, book, confirm, and pay — not just list availability. That means payment gateway integration and real-time confirmation, done in a way that matches the OTA's own booking flow.

Direct connect, not batch

Expedia's direct-connect model expects real-time interaction, not overnight batch updates. Rate changes, last-minute availability, and cancellations all need to reflect instantly — otherwise you risk overbooking or losing the rate parity that protects your margins.

3. A Framework for OTA Integration That Scales

The projects that succeed treat OTA integration as infrastructure, not a one-off feed export. In our experience, four principles make the difference:

Build a translation layer, not a point-to-point patch

Rather than wiring each supplier directly to Expedia's format, build a service-oriented API layer that consolidates data from every source into one internal standard, then translates outward to whatever each OTA requires. This means adding a new supplier — or a new OTA — later doesn't mean rebuilding the integration.

Model it as services, not a monolith

Break the integration into discrete services — search, book, rate update, inventory update — that can run and scale independently. This keeps a spike in one function (say, rate updates during a pricing event) from degrading everything else.

Design for real-time from day one

Batch updates might work for a pilot, but they don't hold up under an OTA's direct-connect requirements. Building for real-time sync from the start avoids a costly re-architecture later.

Test against live conditions before launch

Where possible, validate the integration against real properties and live OTA environments before full rollout. Issues that only appear under real booking traffic — timing conflicts, edge-case rate rules, payment failures — are far cheaper to catch before go-live than after.

4. From APIs to MCP Servers: The Next Shift in Integration

Everything in the framework above still holds — but the way it gets implemented is changing. For the past two decades, “integration” has meant point-to-point APIs: a custom connector built for every pair of systems that need to talk to each other. Your PMS talks to your CRS through one connector. Your CRS talks to Expedia through another. Your payment gateway talks to your booking engine through a third. Each one has its own auth model, its own data shape, its own quirks — and its own maintenance burden.

That approach scales badly. Every new system or channel you add doesn't create one new connection — it creates a new connection to every existing system it needs to talk to. The diagram below is what that looks like in practice for a mid-size hospitality group.

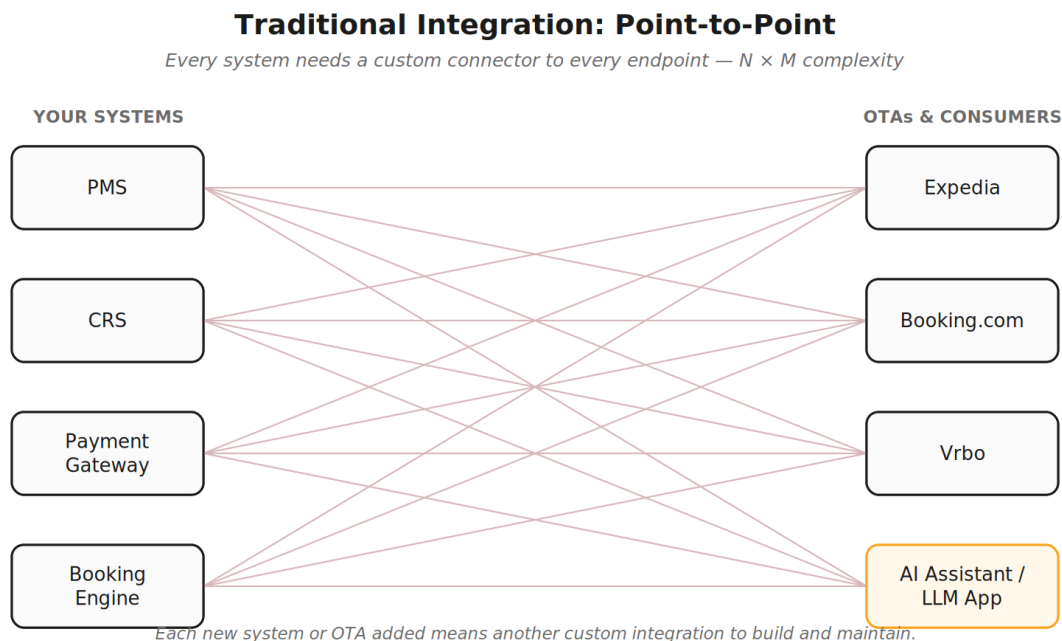


Figure 1. Traditional point-to-point integration: complexity grows with every system and every endpoint.

The Model Context Protocol (MCP) is changing this. MCP is an open standard for connecting systems — and, critically, AI agents — to data and tools through one consistent interface, instead of a custom connector per pair. A system exposes its capabilities once, through an MCP server, and anything that speaks MCP can use them: another internal system, an OTA-facing service, or an AI agent acting on a guest's or an operator's behalf.

Why this matters for hospitality specifically

- Add a system once, not once per connection — a new PMS or a new OTA channel plugs into the existing MCP server rather than triggering a new round of custom integration work
- AI agents get the same access as your other systems — a booking assistant, a revenue-management copilot, or a guest-facing chat agent can query availability or trigger a booking through the same standardized interface, with no separate integration built just for it
- Maintenance shrinks — one protocol to secure, version, and monitor, instead of a sprawling set of bespoke connectors each with their own failure modes
- You're ready for what AI-driven distribution looks like next — as more booking activity gets initiated by AI agents rather than a person clicking through a website, systems that already speak MCP won't need to be re-architected to participate

The shift isn't about replacing the OTA integration work described in this playbook — direct-connect requirements, real-time sync, and payment handling are still exactly as necessary. What changes is the layer underneath: instead of a new custom connector for every system and every consumer, you build to one standardized interface once.

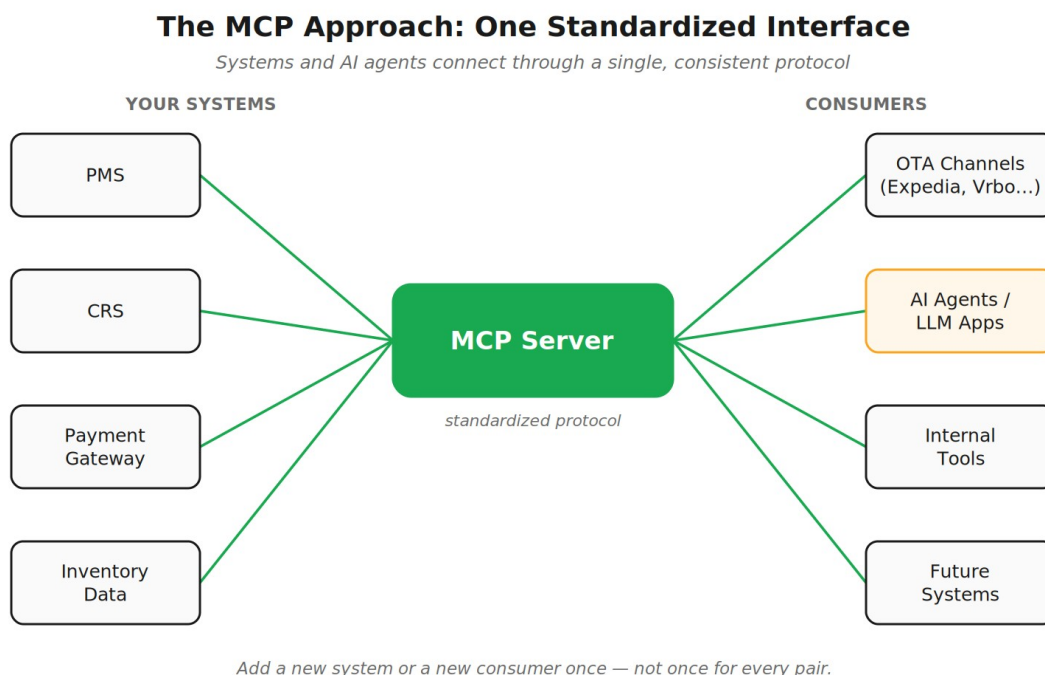


Figure 2. The MCP approach: systems and AI agents connect through a single, standardized server.

Where Keleno fits in

Keleno has been building the kind of API and service-oriented integration work described in this playbook for years — across PMS, CRS, payment, and OTA systems in hospitality. That same team is now applying it to MCP server development: designing and building MCP servers that expose a hospitality operator's systems — inventory, rates, bookings, guest data — through a standardized interface that both existing systems and AI agents can use.

For operators already running an OTA integration, this is a natural next step rather than a rebuild: the underlying services (search, book, rate update, inventory sync) map cleanly onto MCP tools, meaning existing integration investment carries forward instead of being discarded.

5. Case Study: Scaling OTA Distribution for a Top-3 Travel Booking Platform

“The client gained a direct-connect channel to Expedia with real-time visibility into inventory and pricing — letting them respond to market conditions faster than competitors relying on batch updates.”

The Challenge

A leading online travel platform — among the top 3 globally by gross bookings, with a strong presence in vacation rentals, resorts, RV parks, and campgrounds — wanted to expand distribution by listing inventory on Expedia. The goal: more bookings, higher revenue per sale, and stronger customer retention.

The underlying complexity was significant. The platform pulled data from multiple suppliers and Central Reservation Systems that had grown up in silos, with no unified view of room counts, occupancy, rates, or revenue across properties. Any integration needed to:

- Standardize data from multiple sources into Expedia's strict OTA format
- Support switching between different CRS platforms
- Handle the full booking lifecycle, including confirmation and payment gateway integration
- Meet Expedia's direct-connect requirements for real-time inventory and rate accuracy

The Solution

Keleno's team designed a service-oriented architecture (SOA) with a purpose-built API layer that consolidated incoming supplier data and translated it into Expedia's standard format. Core functions — search, book, rate updates, and inventory sync — were built as independent services, enabling parallel processing and consistent performance under load.

Expedia's real-time testing environment allowed the team to validate against live properties throughout development, catching integration issues before launch rather than after.

The Result

The client gained a direct-connect channel to Expedia with real-time visibility into inventory and pricing — letting them respond to market conditions faster than competitors relying on batch updates. The architecture was built to scale, supporting future OTA integrations beyond the initial rollout.

6. Common Pitfalls to Avoid

- Treating the integration as a single export job instead of ongoing infrastructure that needs monitoring and maintenance
- Underestimating how much internal data cleanup is needed before anything can be standardized
- Building a connection tied to one CRS or one OTA's format, which becomes a rebuild the moment you need to add another
- Skipping real-time testing against live conditions and discovering edge cases only after launch
- Leaving payment and booking-confirmation flows as an afterthought rather than part of the core design

7. Readiness Checklist: Are You Prepared to Integrate?

Before starting an OTA integration project, it's worth honestly assessing where you stand:

- Can you currently produce a single, accurate view of inventory, rates, and occupancy across all your properties?
- Do you know how many separate systems (CRS, PMS, channel managers) your data currently lives in?
- Is your rate and availability data updated in real time, or on a batch/manual cycle?
- Do you have a payment gateway that can support real-time booking confirmation?
- Have you reviewed the specific mandatory fields and formatting rules required by the OTAs you want to list on?
- Do you have a plan for testing against live conditions before full rollout, rather than only in a staging environment?

If several of these are unclear, that's normal — it's exactly the gap an experienced integration partner should help you close before writing any code.

8. About Keleno & Next Steps

Keleno Labs is an IT services and product engineering company specializing in IoT/IIoT, AI/ML, and cloud engineering, with primary focus in Hospitality and Healthcare.

Our hospitality practice has built proven integrations across major OTAs and PMS platforms, and we're helping hospitality customers move toward AI-based solutions — including MCP server development that solves connectivity challenges between hospitality systems and AI agents.

If your organization is considering OTA distribution — or already has an integration that's become difficult to maintain or extend — we'd welcome the conversation.

Let's talk

Email: info@keleno.com

Phone: +91 8848358534

www.keleno.com