

**Learn to
Design Cloud
Architecture**





Home

About

Rooms

Page

Contact

Search

Hotel Booking

Discover Now

Check In

MM/DD/YYYY

Check Out

MM/DD/YYYY

Room

01

Adult

01

Children

01

Book Now

What is a Hotel Booking System?

- A hotel booking system is software that allows guests to reserve rooms online and helps hotels manage reservations and availability.

Booking.com



Expedia

airbnb

The logo for Airbnb, consisting of a red abstract house-like icon followed by the word "airbnb" in a lowercase sans-serif font.

agoda

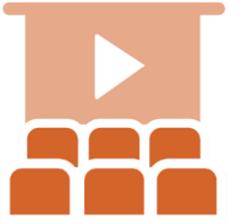


Similar systems



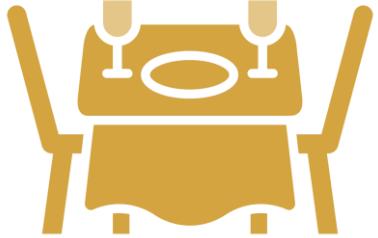
Travel & Transportation

- Hotel booking
- Flight reservation
- Parking lot reservation
- Car rentals



Entertainment & Events

- Movie ticket reservation
- Concerts and event ticketing
- Tour or activity booking



Food & Hospitality

- Restaurant reservations
- Festival or fair entry passes
- Lounge table reservations



Health & Wellness

- Doctor or clinic appointments
- Salon or spa appointments
- Gym class bookings

Airbnb - Holiday property rentals company

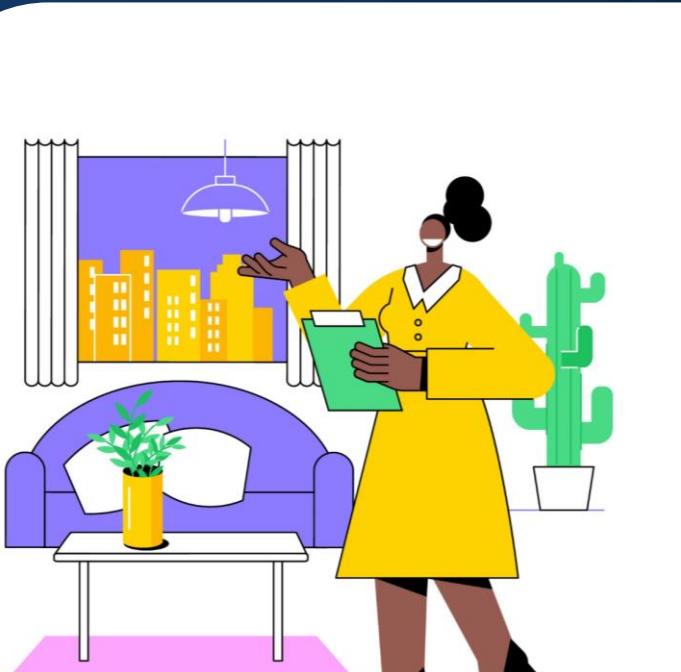
- Original Idea:
 - Started as "AirBed & Breakfast"
- User Base:
 - Over 7 million properties worldwide , Over 150 million users globally
- Business Model:
 - Charges service fees to both hosts and guests



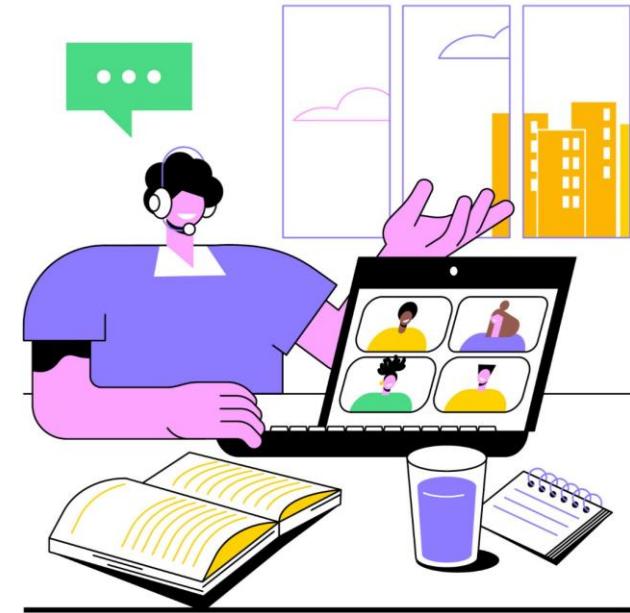
User Profiles



Guest

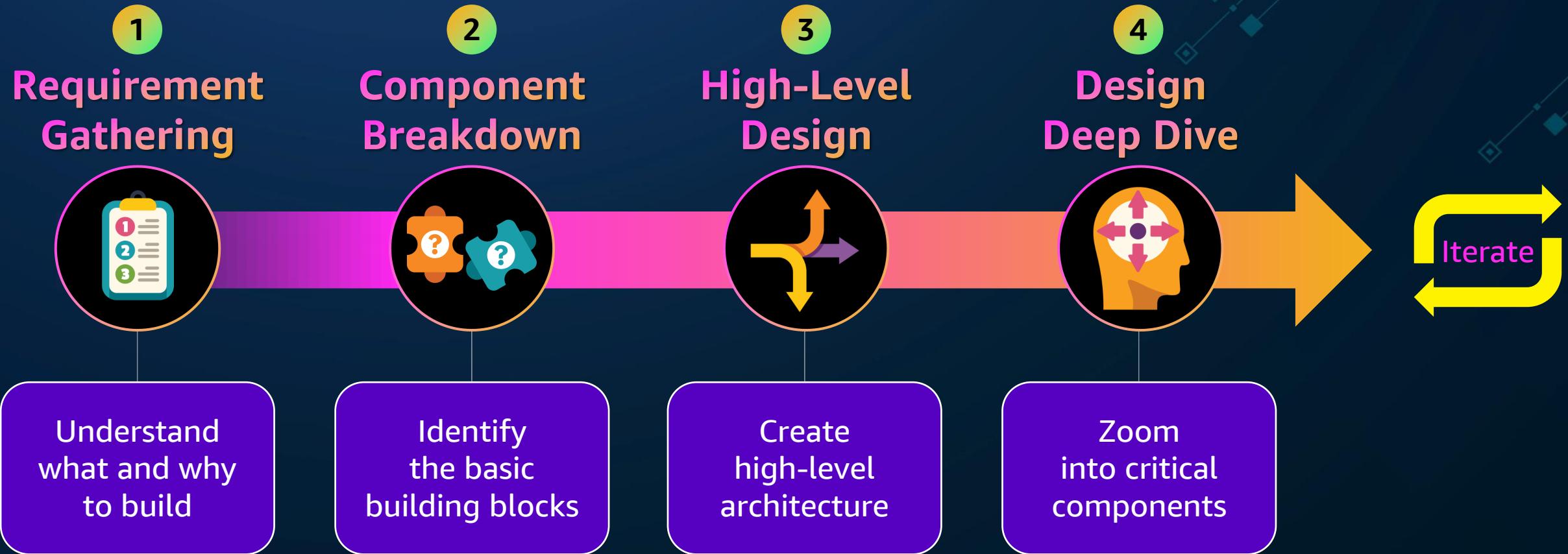


Host



Ops.

Designing a System – A simple framework



Requirement Gathering

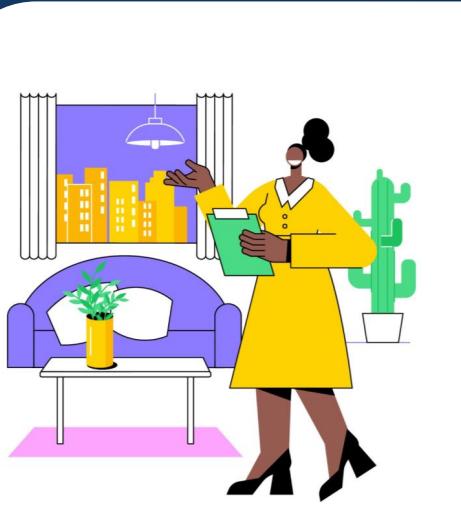


Functional requirement from a booking system



Guest

- Search and view listings
- Submit booking request and payment
- Modify booking requests



Host

- List and update property
- Handle bookings (accept/reject or automate)
- Cancel a booking after accepting



Ops.

- Validate and approve listing requests
- Communicate with Guest and Host for dispute mediation, refunds, suggesting alternatives

Non-functional requirement

Strong Consistency for Bookings and Availability

1

Low latency property search

3

Security and Privacy

5

2

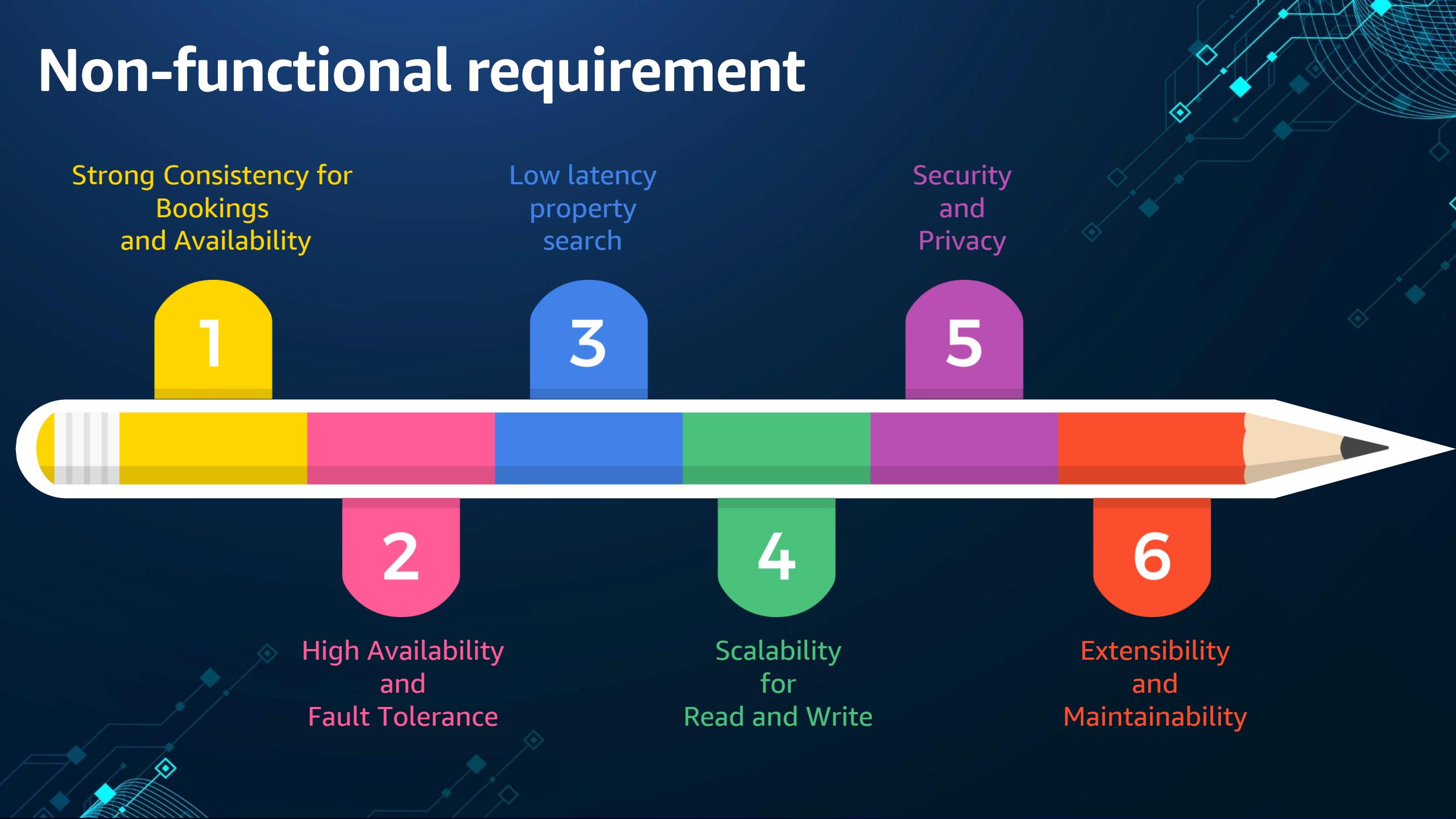
High Availability and Fault Tolerance

4

Scalability for Read and Write

6

Extensibility and Maintainability



Out of scope



- Authentication
- Analytics
- Dynamic Pricing
- Payment integration
- Value added services
- Property/Guest Reviews
- Customer Support
- Notification Services
- Archival Service
- Localization



Component Breakdown



Component Breakdown



User
Interface



Property
Listing Service



Search
Service



Property
Reservation Service



Reservation
Management Service



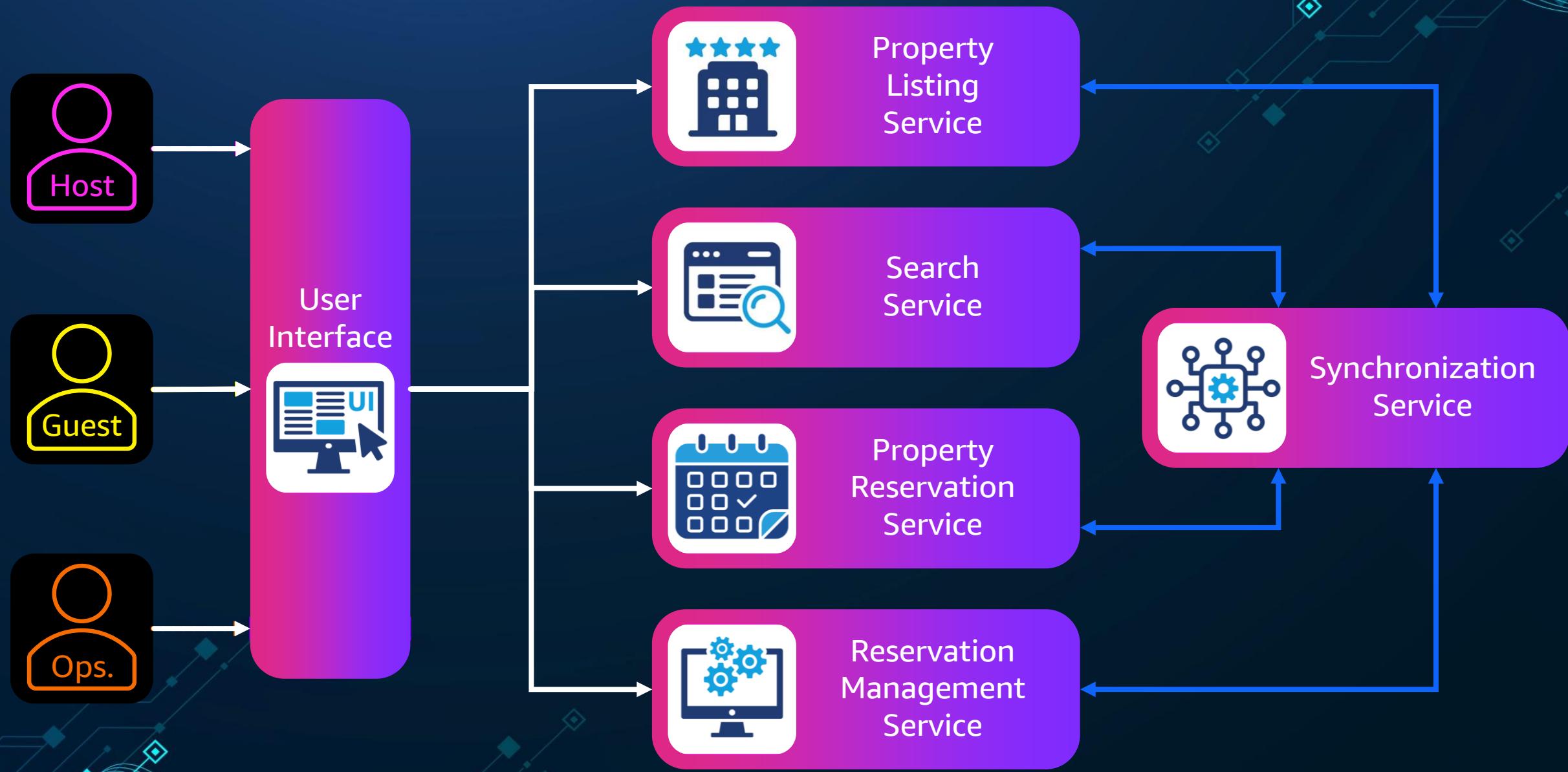
Synchronization
Service



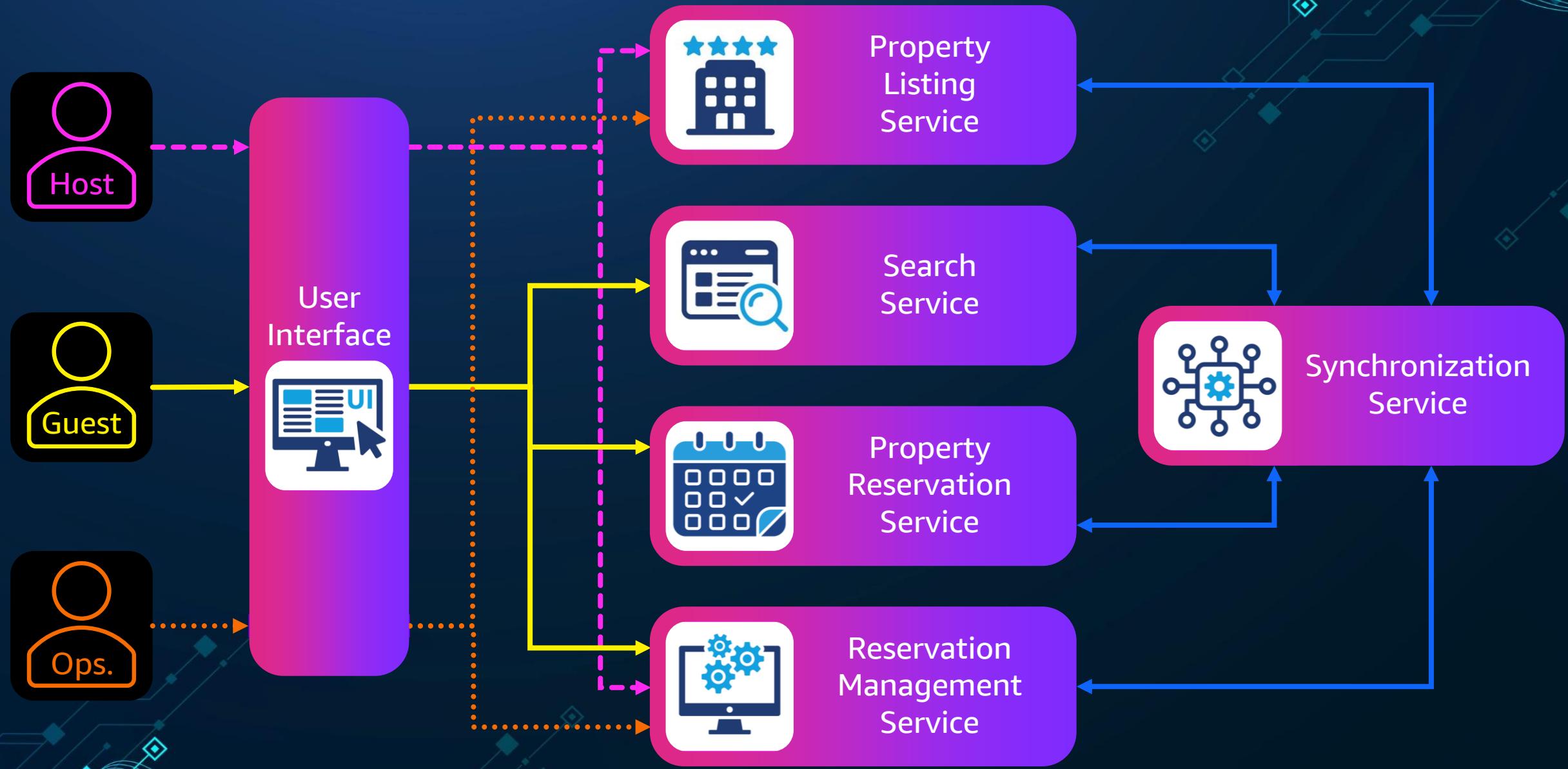
High-Level Design



Simplified High Level Design



High Level Design

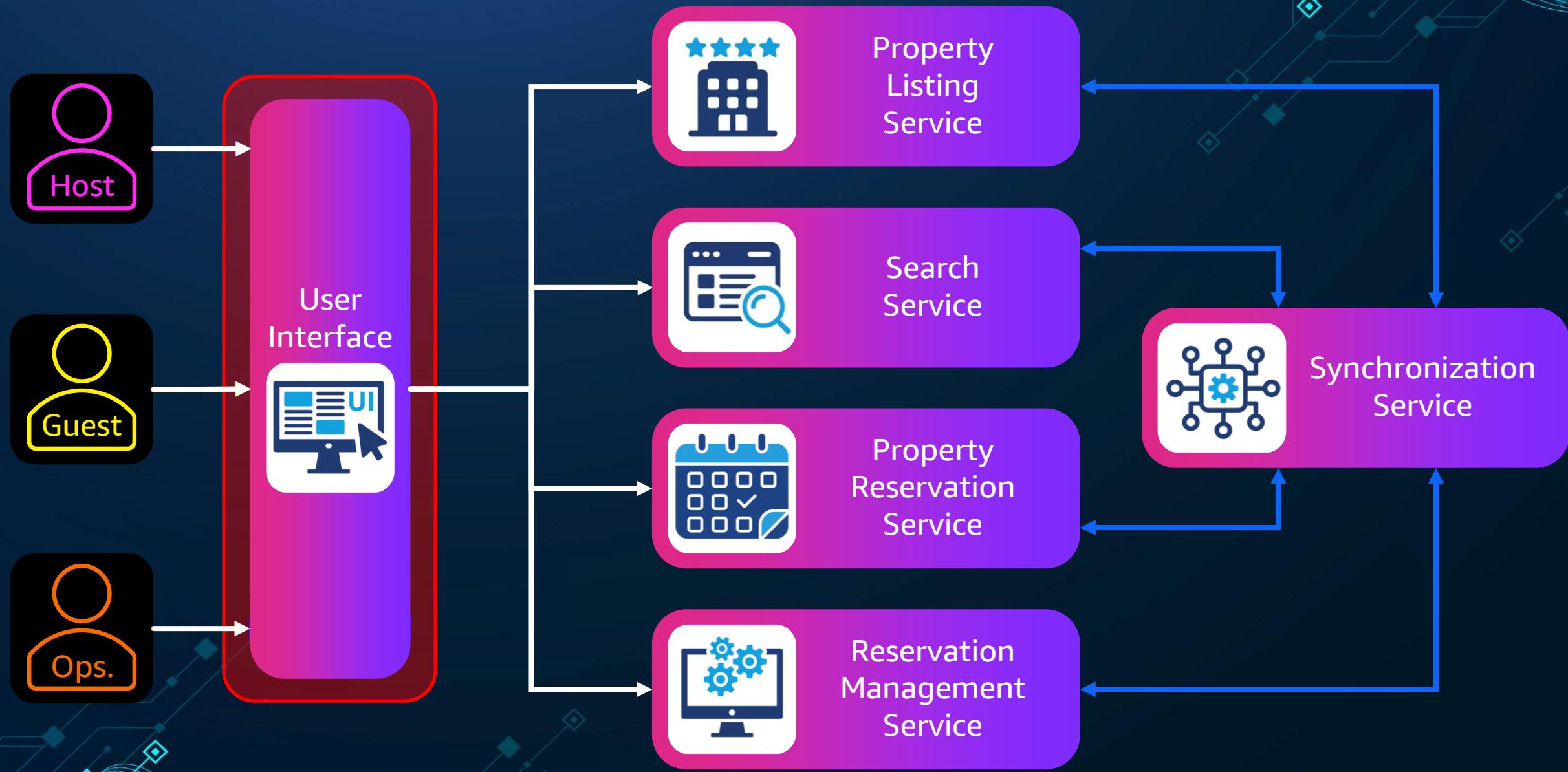




Design Deep Dive



Simplified High Level Design



Simplified High Level Design



Security & Performance Considerations



- HTTPS-only communication
- Role-based view control
- Secure session/token handling
- Bot protection (e.g., CAPTCHA)
- CDN-delivered static assets
- Fast-rendering UI with caching

Frontend Presentation Layer



- Role-based UI experience (Guest, Host, Admin)
- Consistent layout and branding
- Responsive design for mobile/desktop
- Internationalization (i18n) support

Access & Identity Layer



- Centralized authentication and session management
- Secure login/registration
- Role-based access enforcement
- User identity passed along with each request

Navigation & Interaction Layer



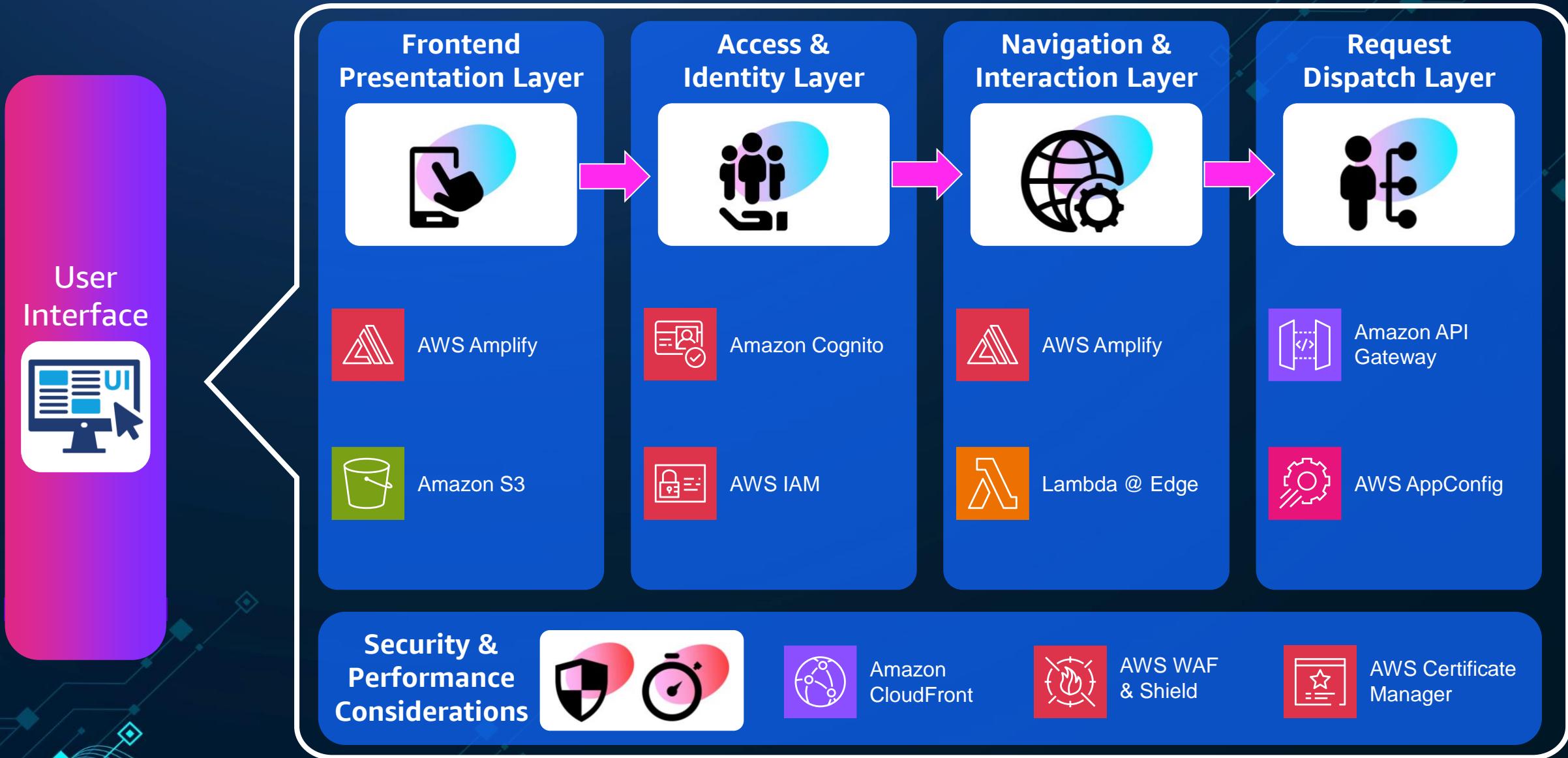
- Public and protected pages
- Dynamic routes for listings and bookings
- Dashboards tailored by role
- Routing decisions based on user type and permissions

Request Dispatch Layer

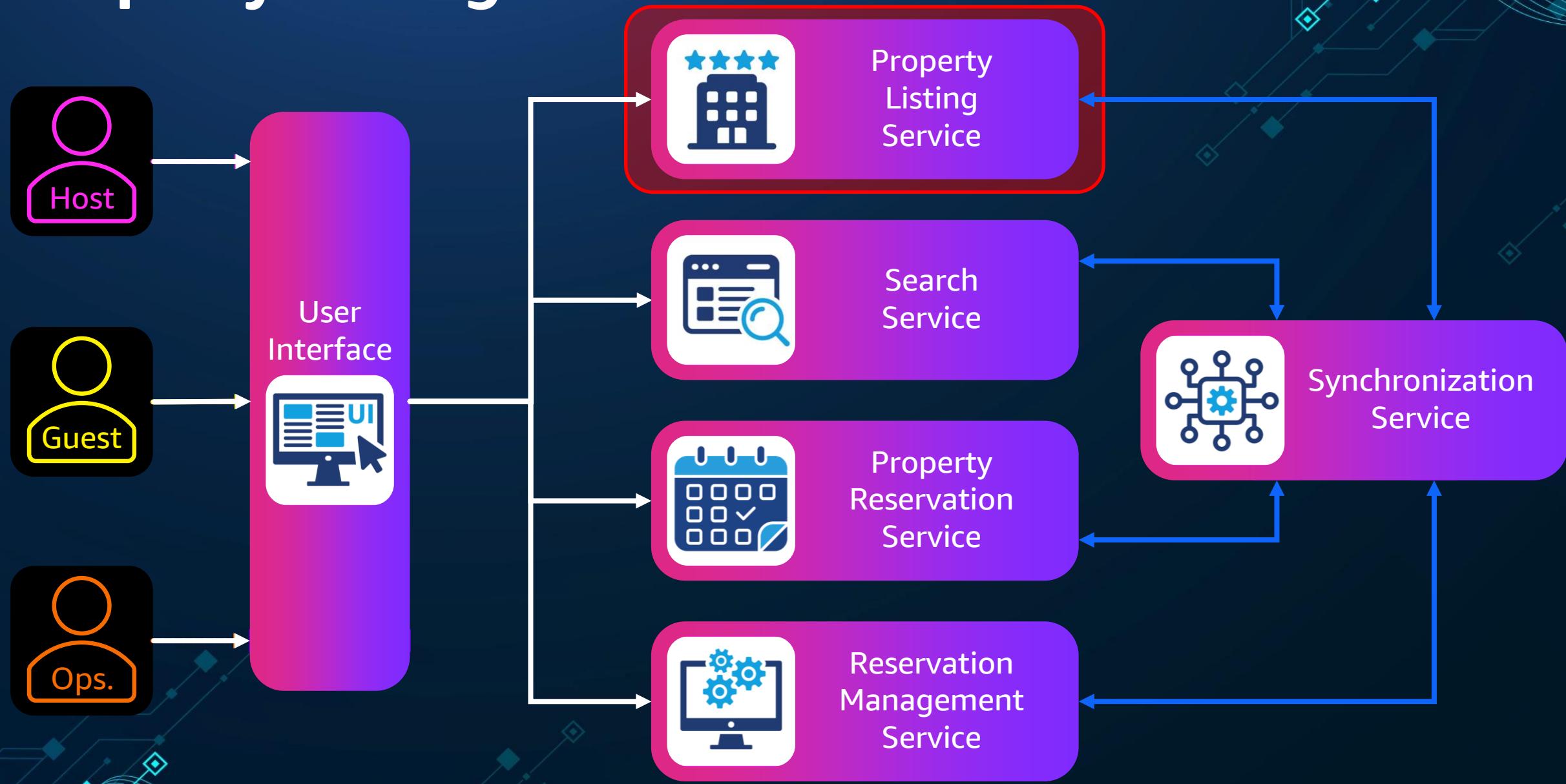


- All frontend requests flow through a Backend for Frontend (BFF) or API Gateway
- BFF handles:
 - Authentication verification
 - Role enforcement
 - Input validation
 - Forwarding requests to correct backend service

Implementation on AWS



Property Listing Service

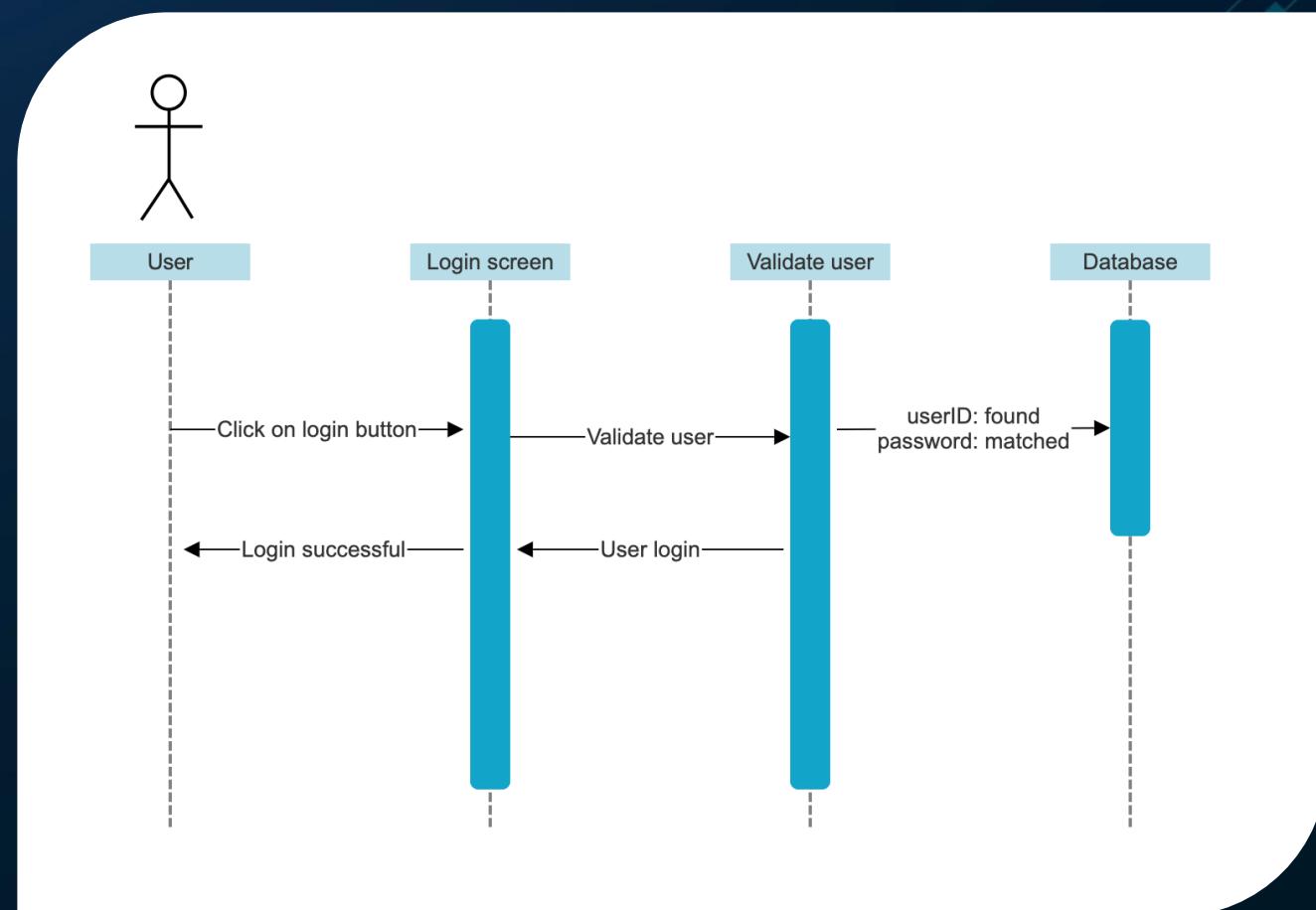


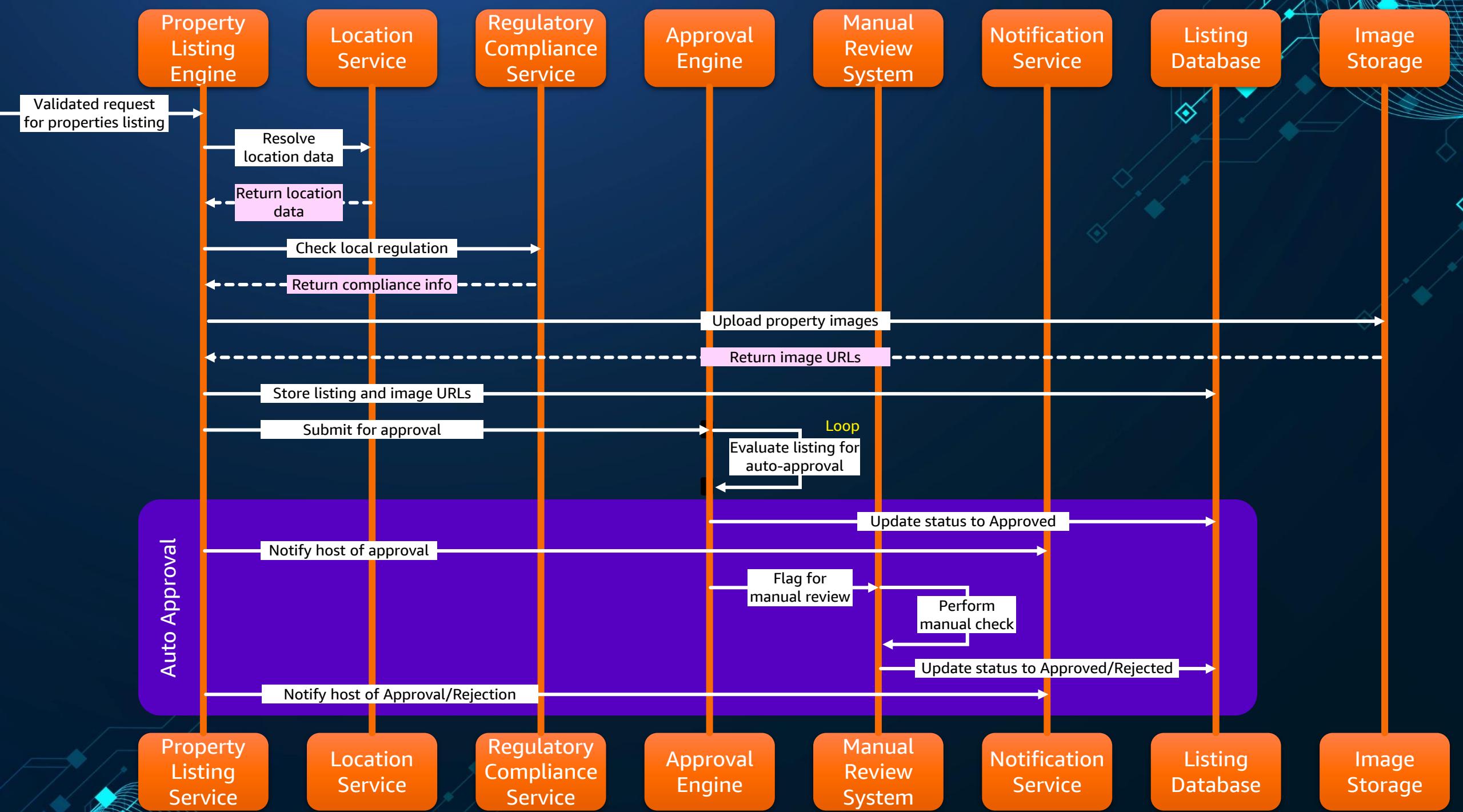
Property Listing Service



Sequence Diagram

- A sequence diagram is used in software engineering and systems design to show how objects or components interact with each other.
- Helps you visualize how and in what order different parts of a system interact.
- Some common tools:
 - [Lucidchart](#)
 - [Draw.io](#)
 - [SequenceDiagram.org](#)
 - [Eraser.io](#)





Implementation on AWS



Property Listing Engine



Amazon API Gateway



AWS Lambda



AWS Step Functions



Manual Review System



AWS Step Functions



Location Service



Amazon Location Service



Notification Service



Amazon SNS



Amazon EventBridge



Regulatory Compliance Service



AWS Lambda



Amazon S3



Listing Database



Amazon Aurora



Amazon OpenSearch Service



Approval Engine



AWS Lambda



Amazon SQS



Image Storage



Amazon S3



Amazon CloudFront

Leveraging AWS AI/ML Services



Amazon Rekognition

Image analysis
and
moderation

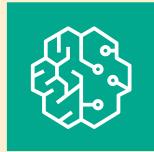
- Detect inappropriate or unsafe content (e.g., nudity, violence, explicit signs).
- Label property features (e.g., kitchen, bathroom, pool).
- Facial blurring (if needed for privacy).



Amazon Comprehend

Text analysis of
property
descriptions

- Detect profanity or policy violations.
- Extract location or entity mentions (e.g., city, beach, mountain).
- Sentiment analysis
 - Optional for evaluating tone.



Amazon SageMaker

Custom models
for
listing approval

- Train a model to auto-approve listings based on historical approvals.
- Anomaly detection on pricing, location, or image patterns.
- Ranking listings based on predicted popularity or quality.



Amazon Textract

Extract info from
regulatory documents
uploaded by hosts

- Parse PDFs like permits, licenses, and compliance forms.
- Extract structured fields for downstream validation or audits.

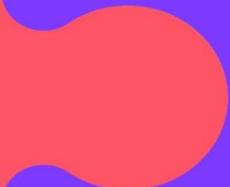
Integration ideas



Use **Rekognition** and **Comprehend** in your approval workflow via AWS Lambda + Step Functions.



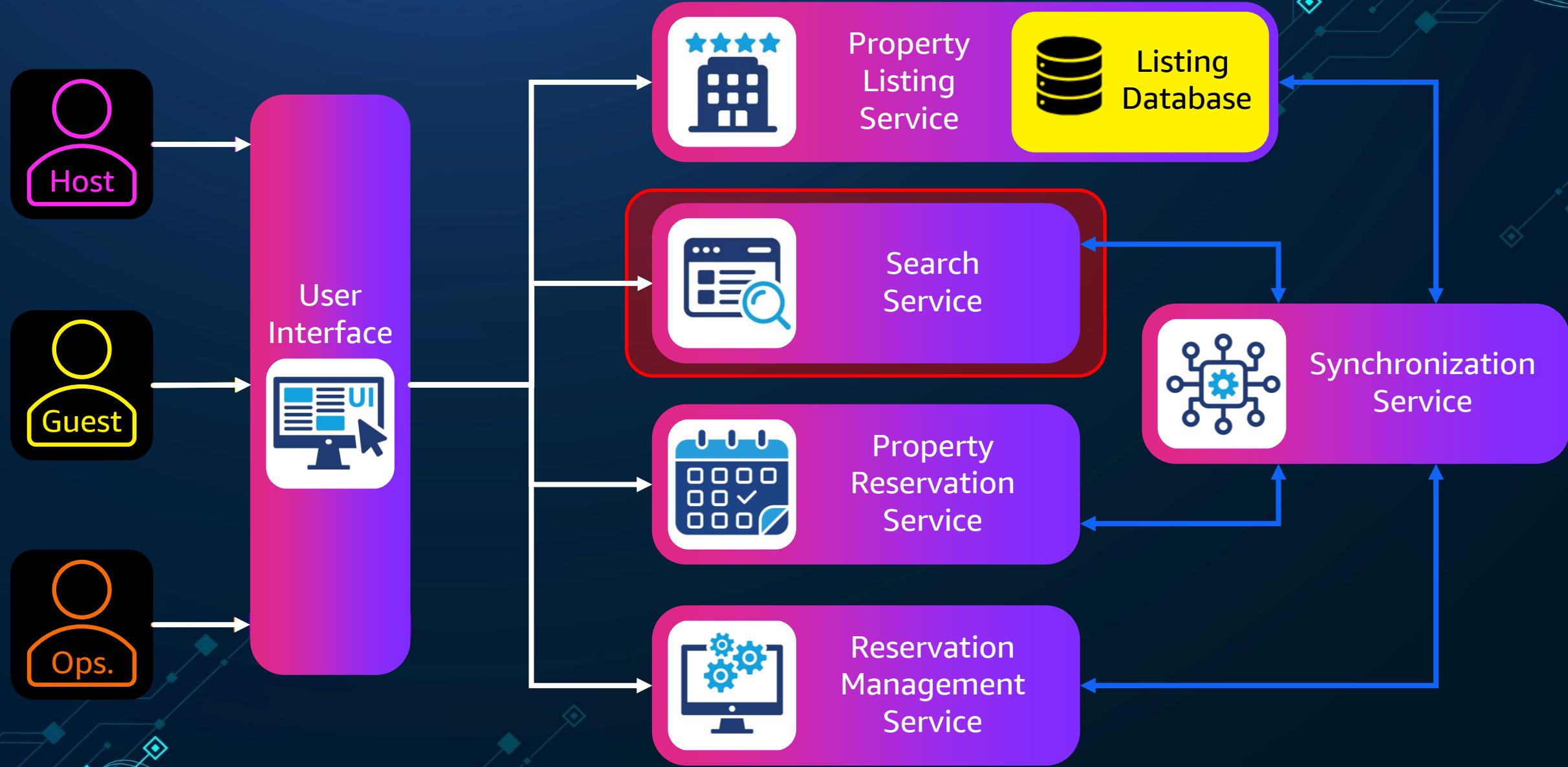
Feed rejected listings to **SageMaker** for training improved auto-approval models.



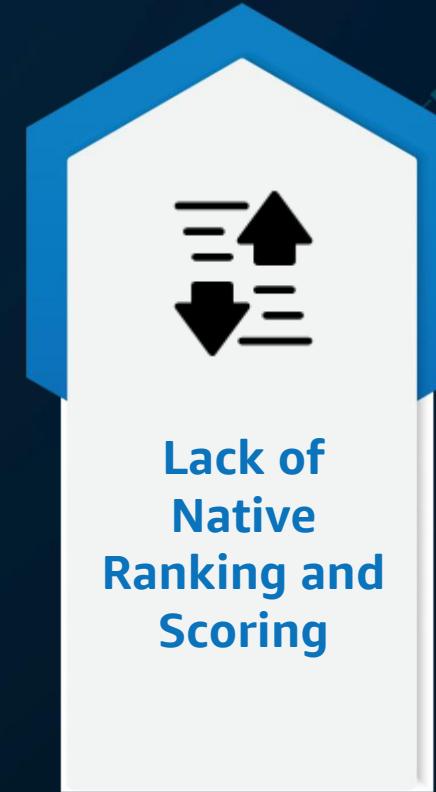
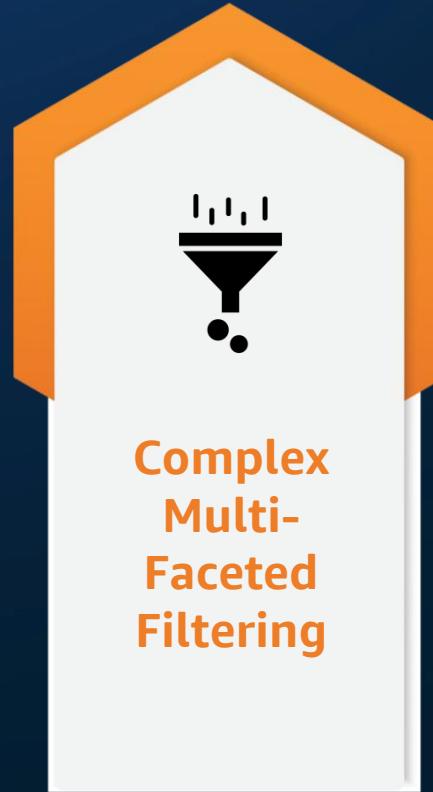
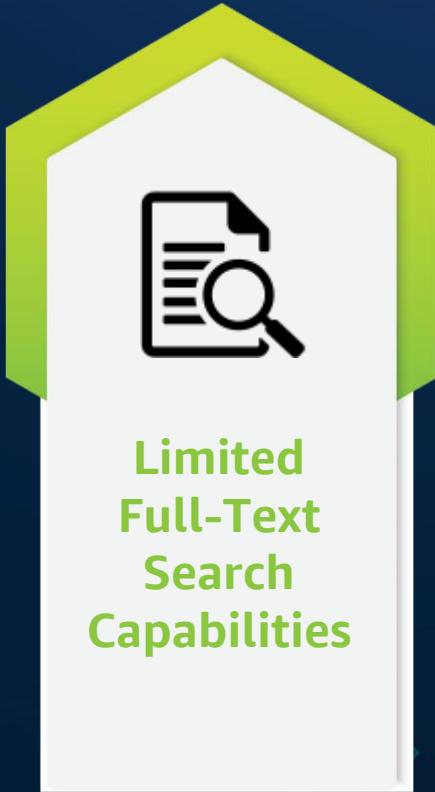
Use **Textract** with to automatically verify document fields with regulatory databases.



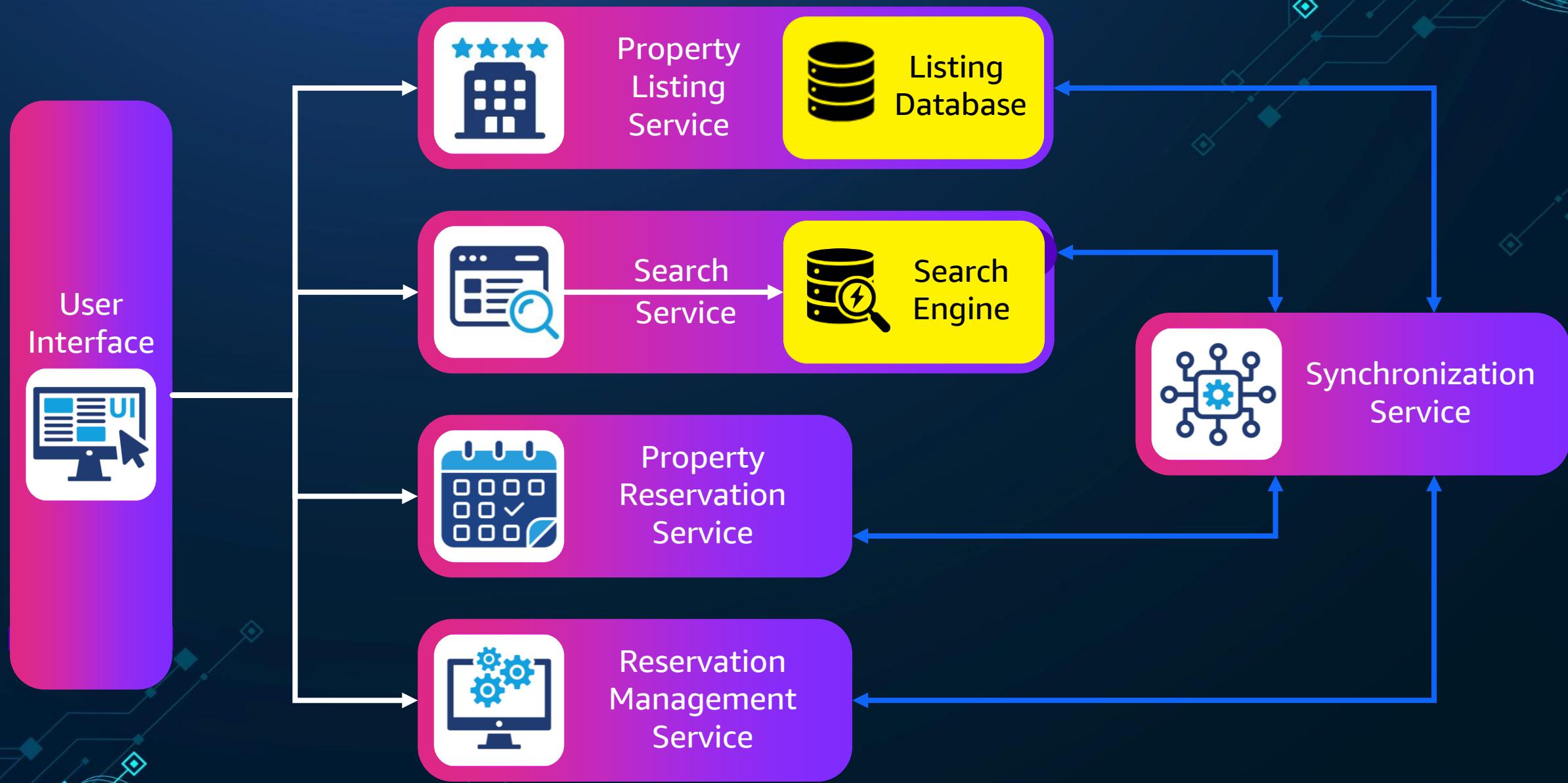
Search Service



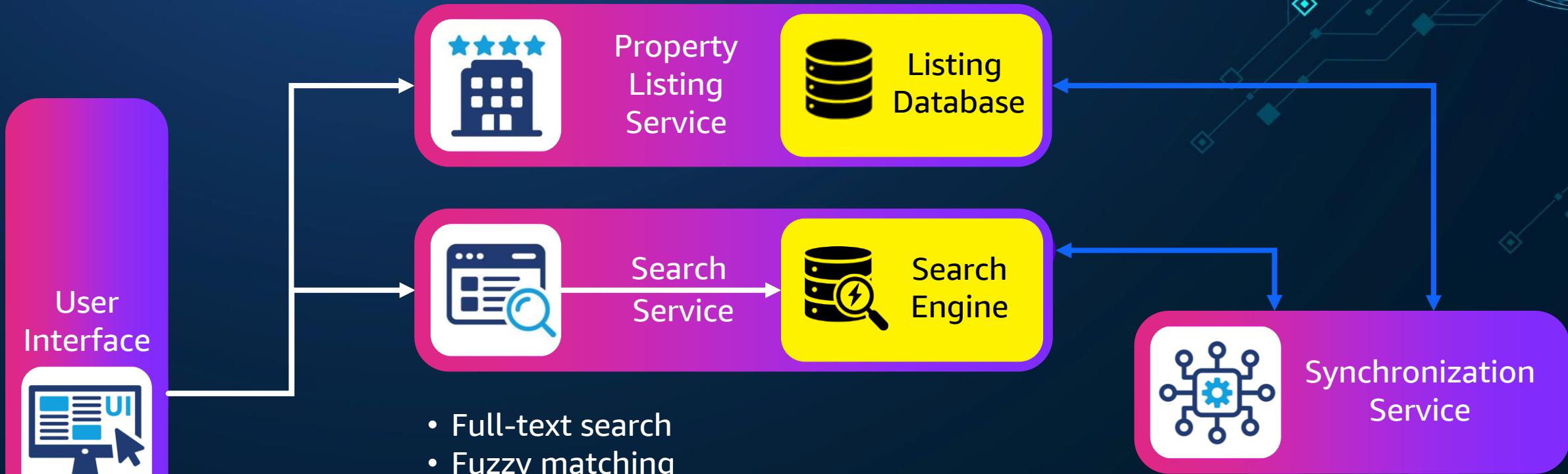
Challenges of searching in Listing Database



Hybrid Architecture: RDBMS + Search Engine



Hybrid Architecture: RDBMS + Search Engine



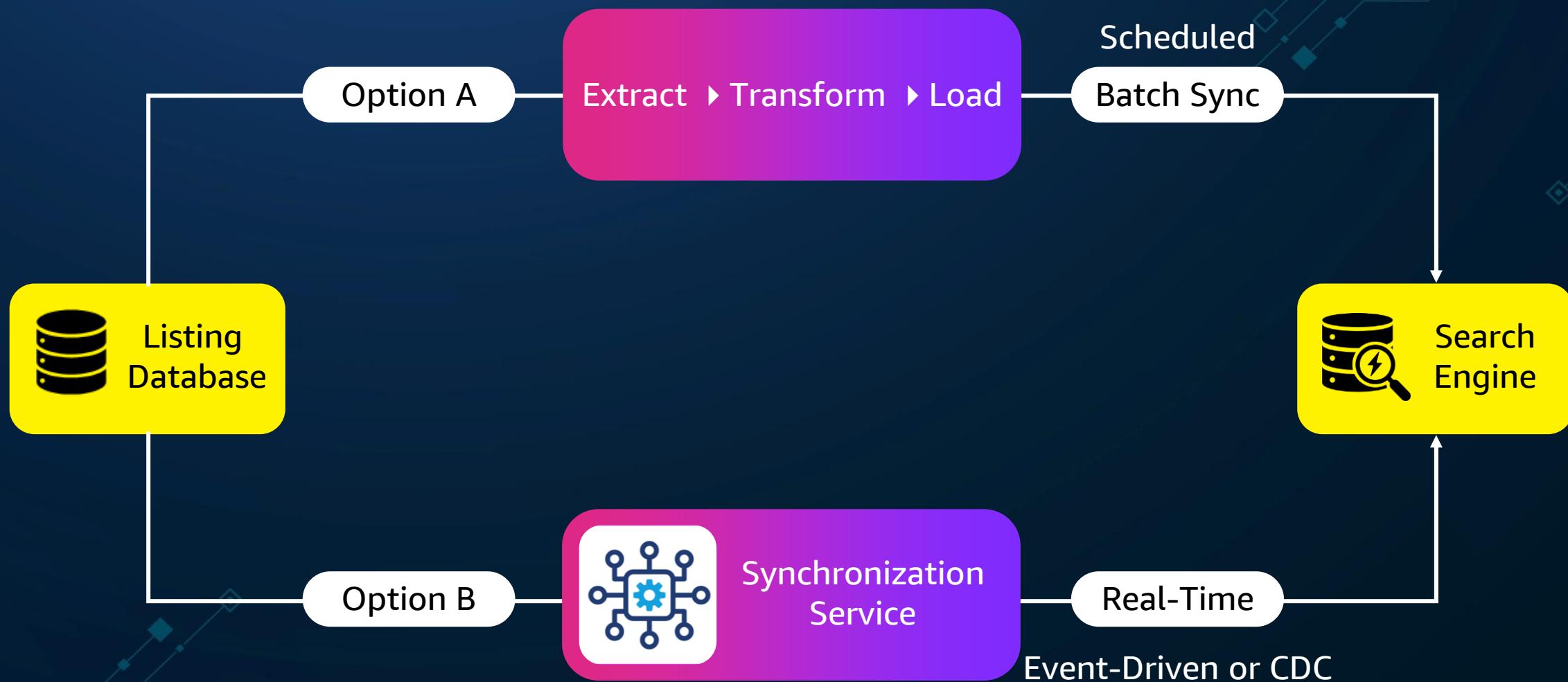
- Full-text search
- Fuzzy matching
- Auto-complete suggestions
- Geo-distance filtering
- Relevance-based ranking



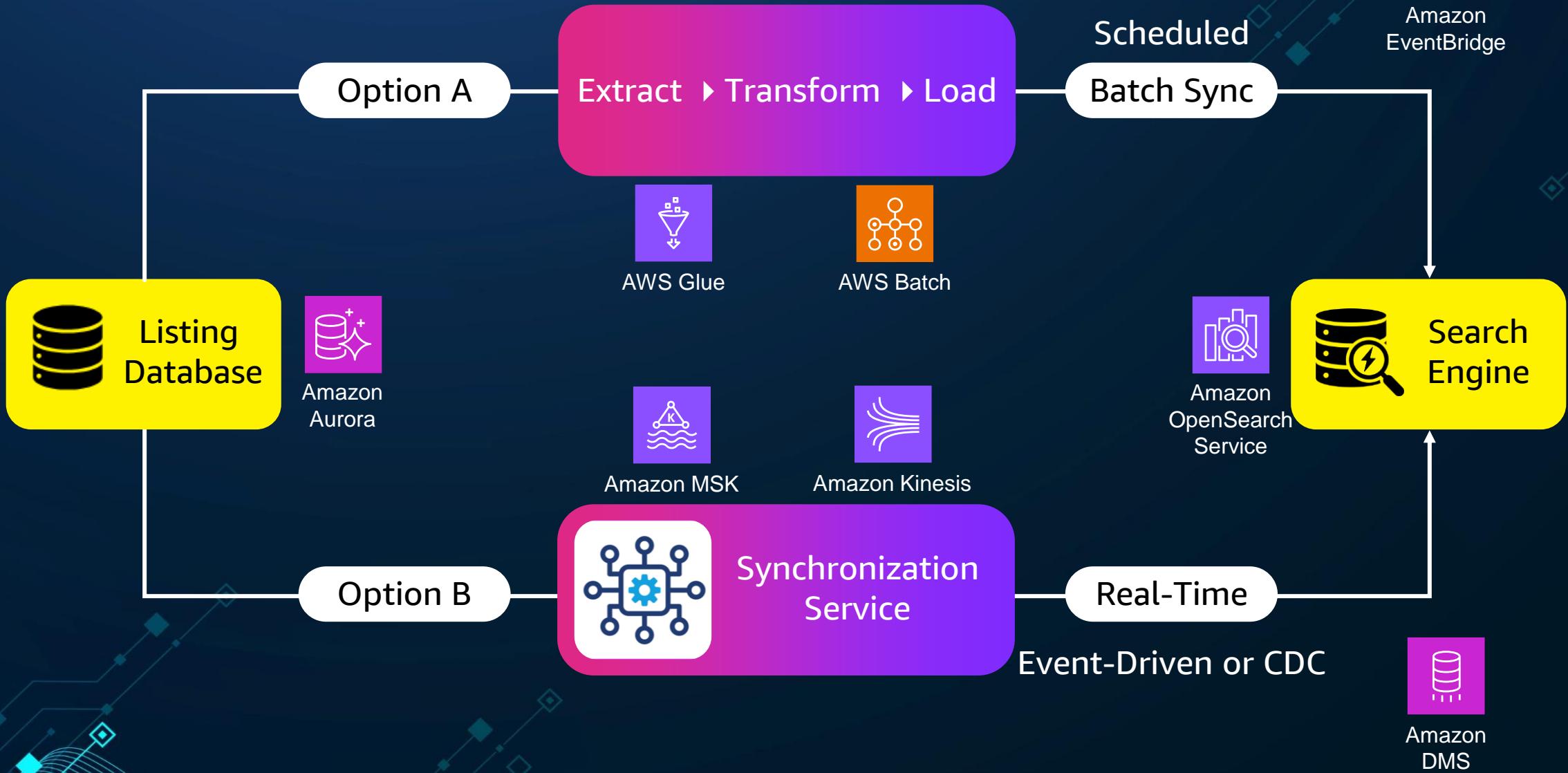
elasticsearch



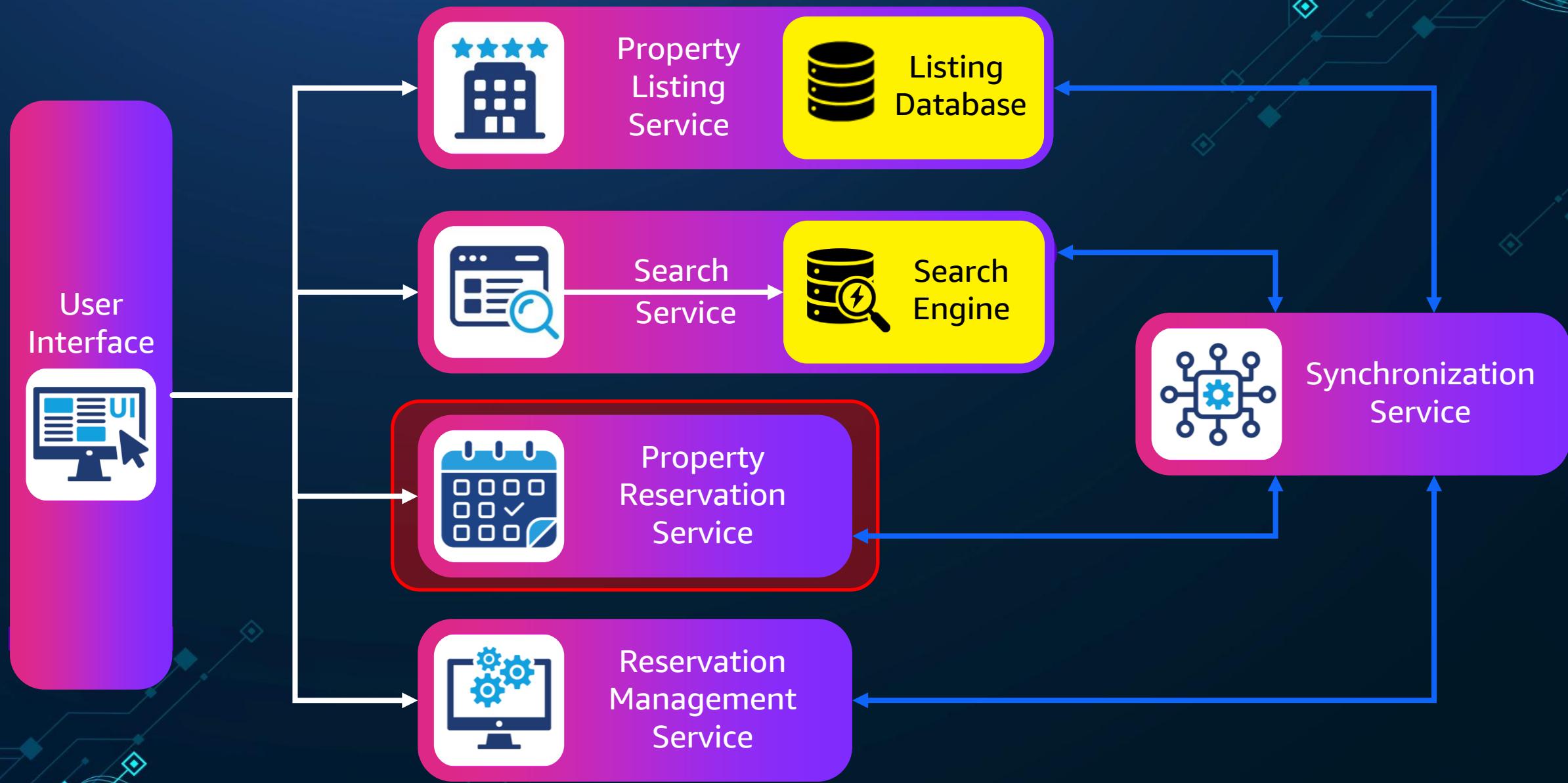
Syncing Data from RDBMS to Elasticsearch



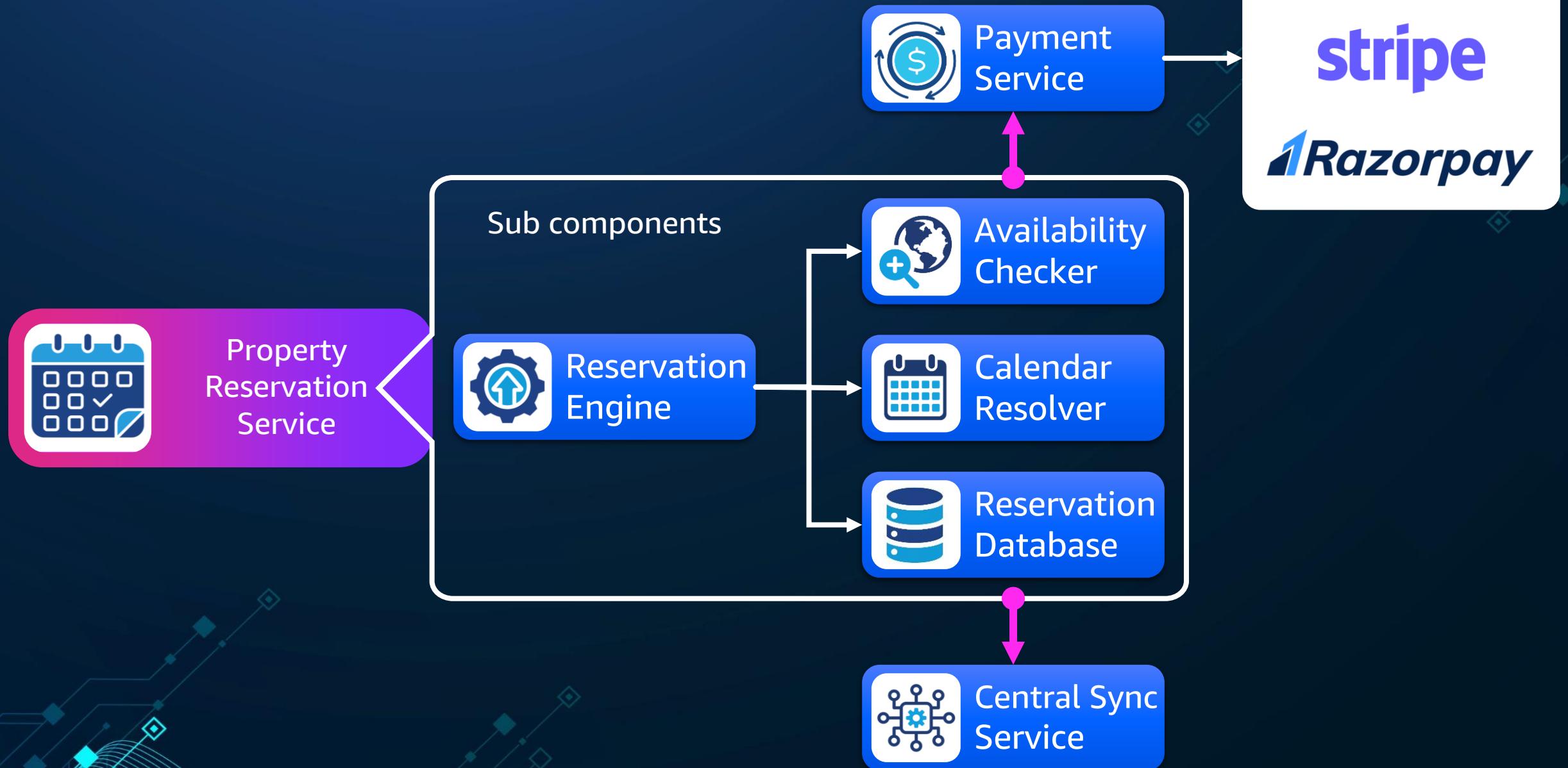
Implementation on AWS

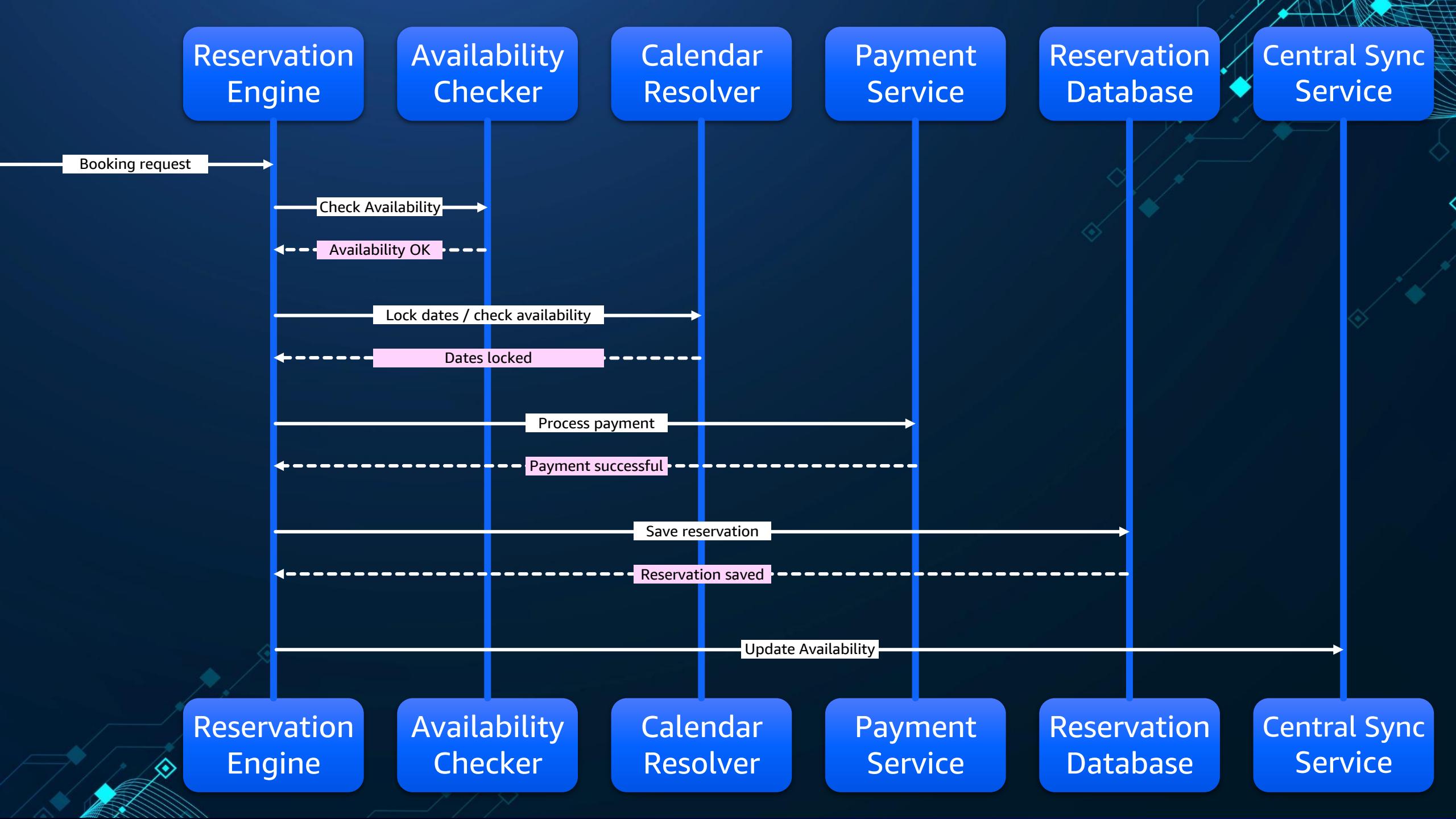


Simplified High Level Design

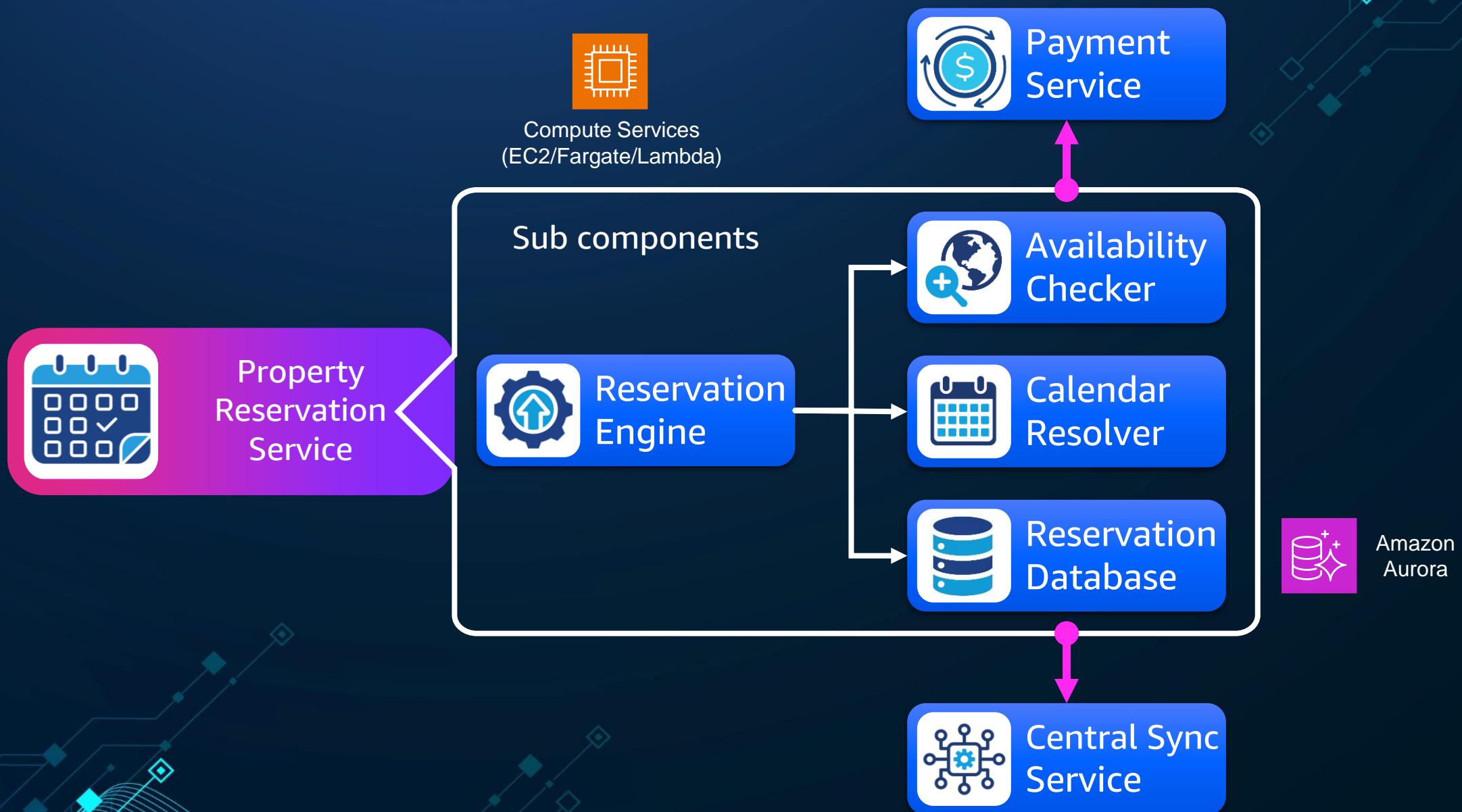


Property Reservation Service

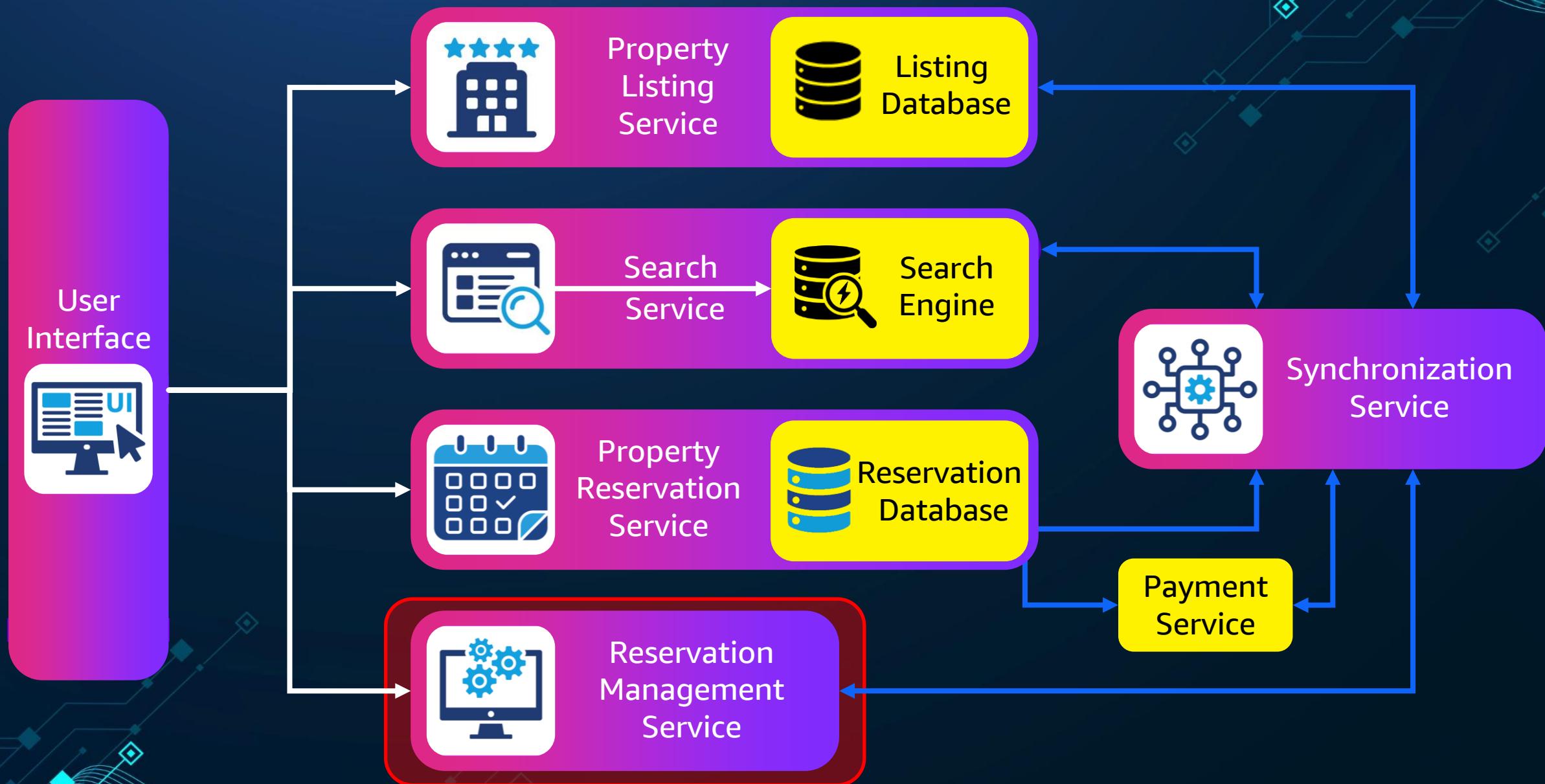




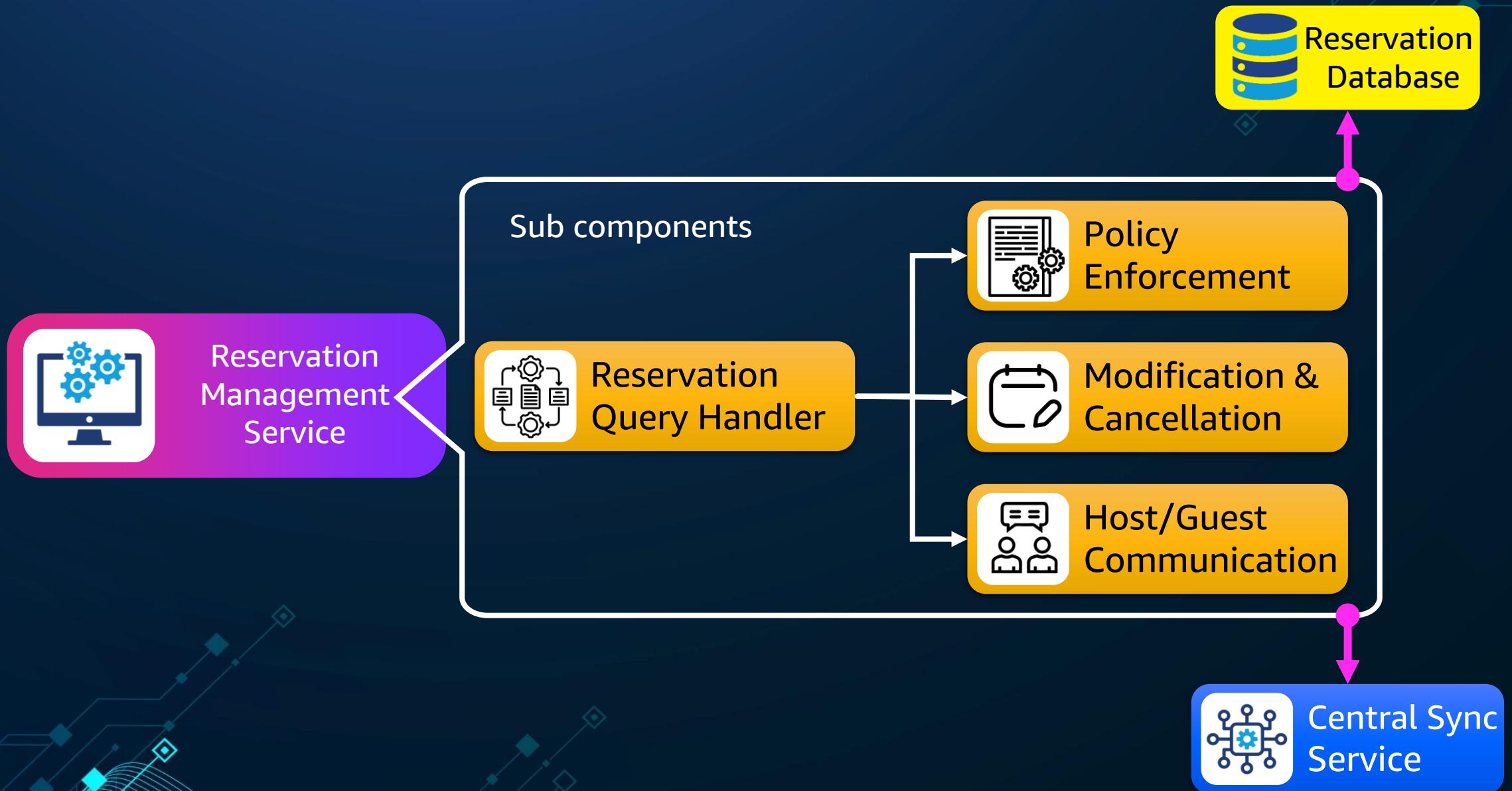
Implementation on AWS



Reservation Management Service



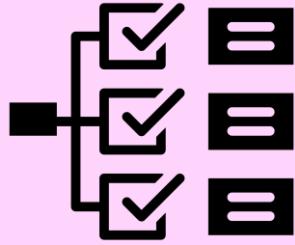
Reservation Management Service



Synchronization Service

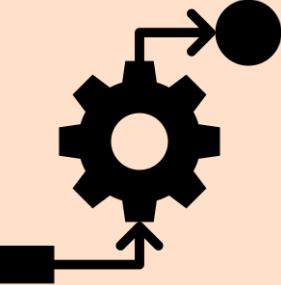


Synchronization Service – Core Functionality



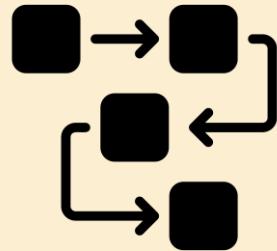
Data Propagation & Sync Across Services

- Keeps the data consistent across various services.
- Example: When a property is booked, the sync service updates the availability in the listing DB and the search index.



Event-Driven Communication

- Publishes and listens to events about property update, reservation confirmation, payment success
- Allows services to remain loosely coupled.



Triggering Downstream Updates

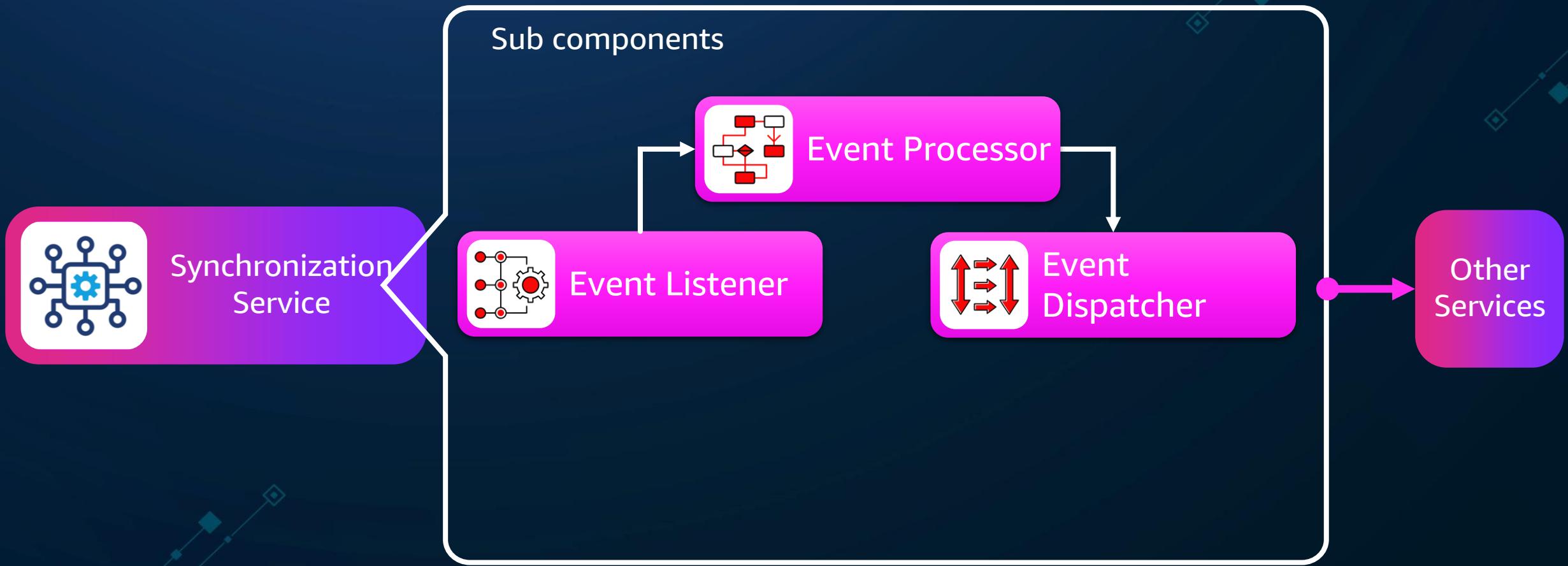
- Initiates further actions or updates in other services.
- For example - Search engine reindexing when listings change.
- Payment updates after reservations.



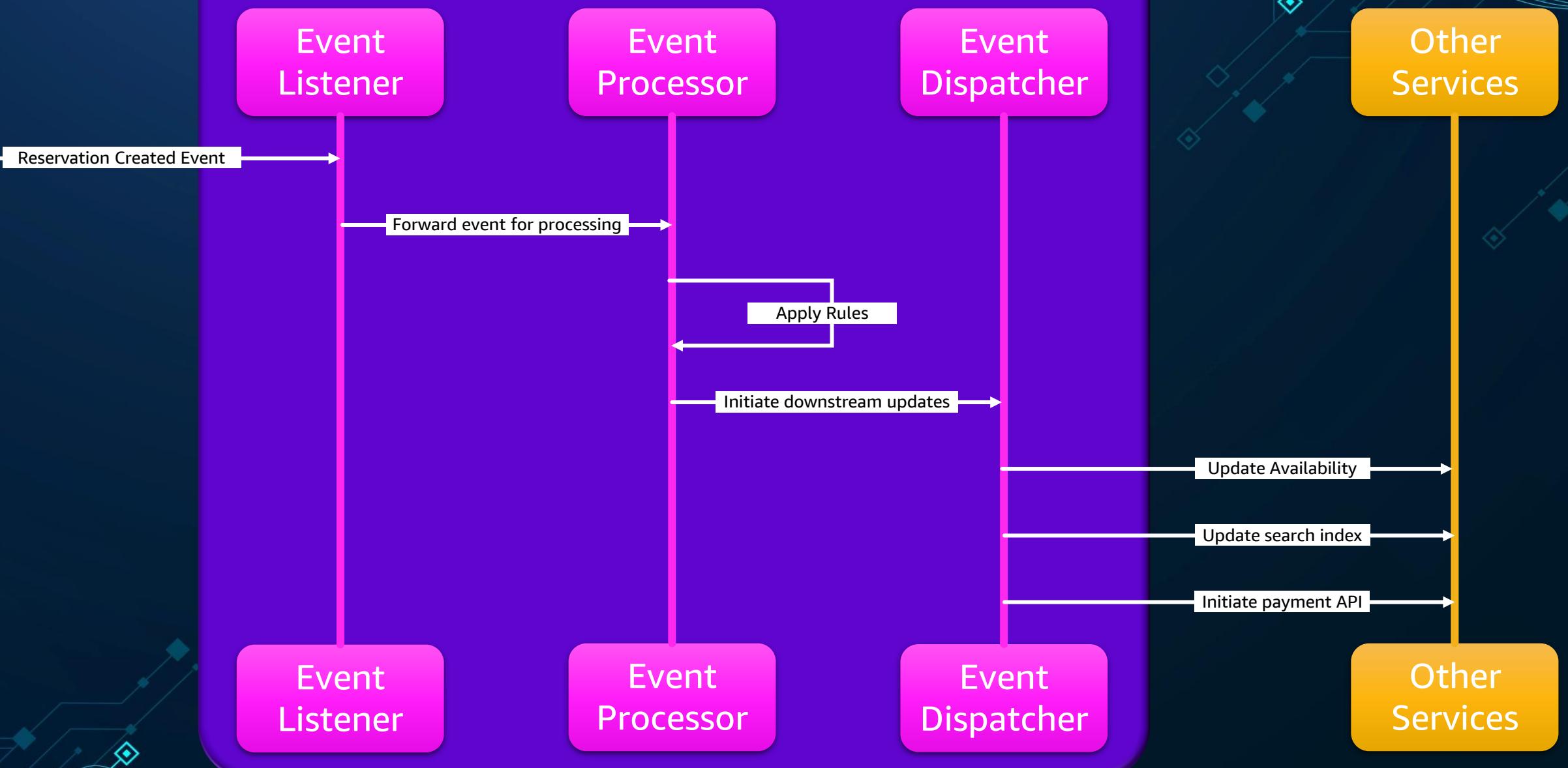
Audit Logging / Change Tracking

- Keeps a record of all the important events or changes, like a journal or audit trail.
- Why it matters:
 - Debugging
 - Auditing
 - Rollbacks

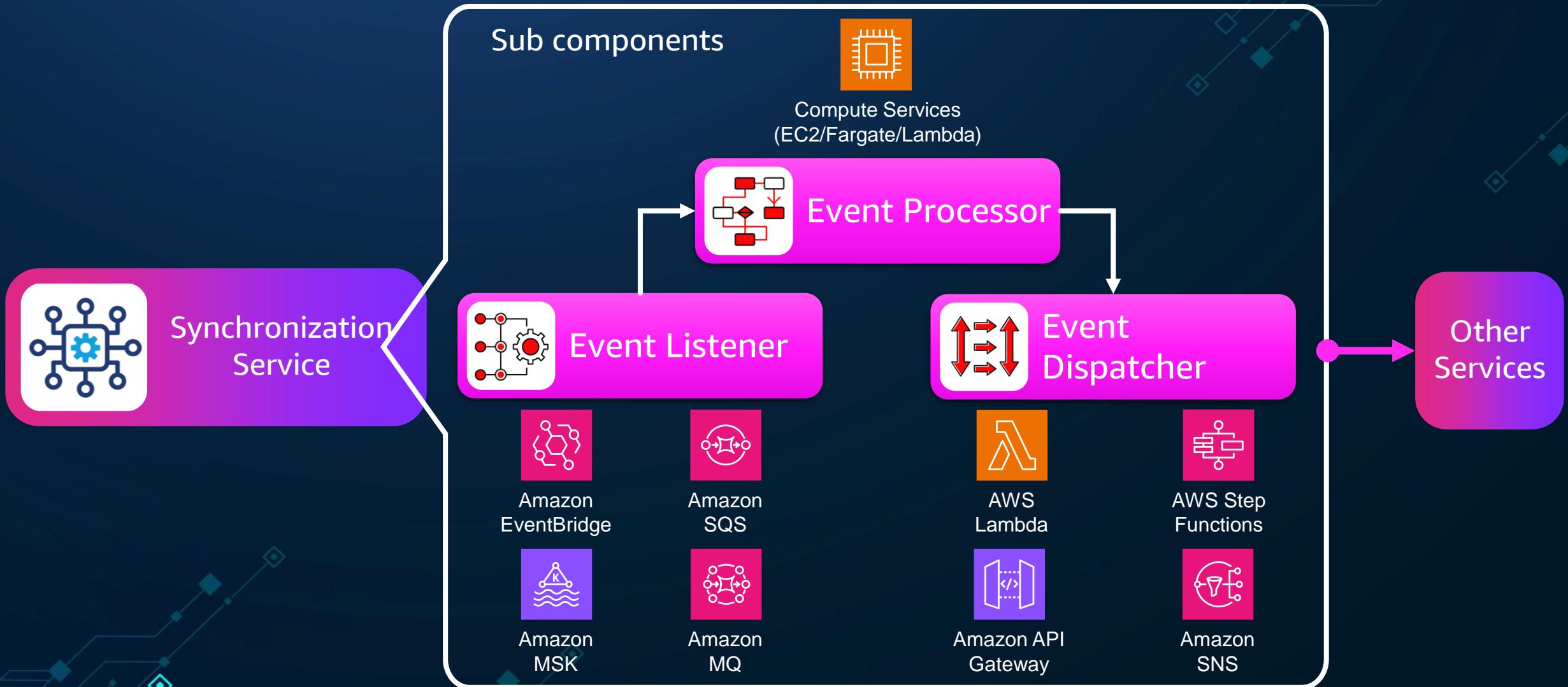
Synchronization Service – Sub components



Synchronization Service



Implementation on AWS



Updated Architecture



AWS Solutions Library

- Guidance for Serverless Reservation System for Lodging on AWS - [Link](#)

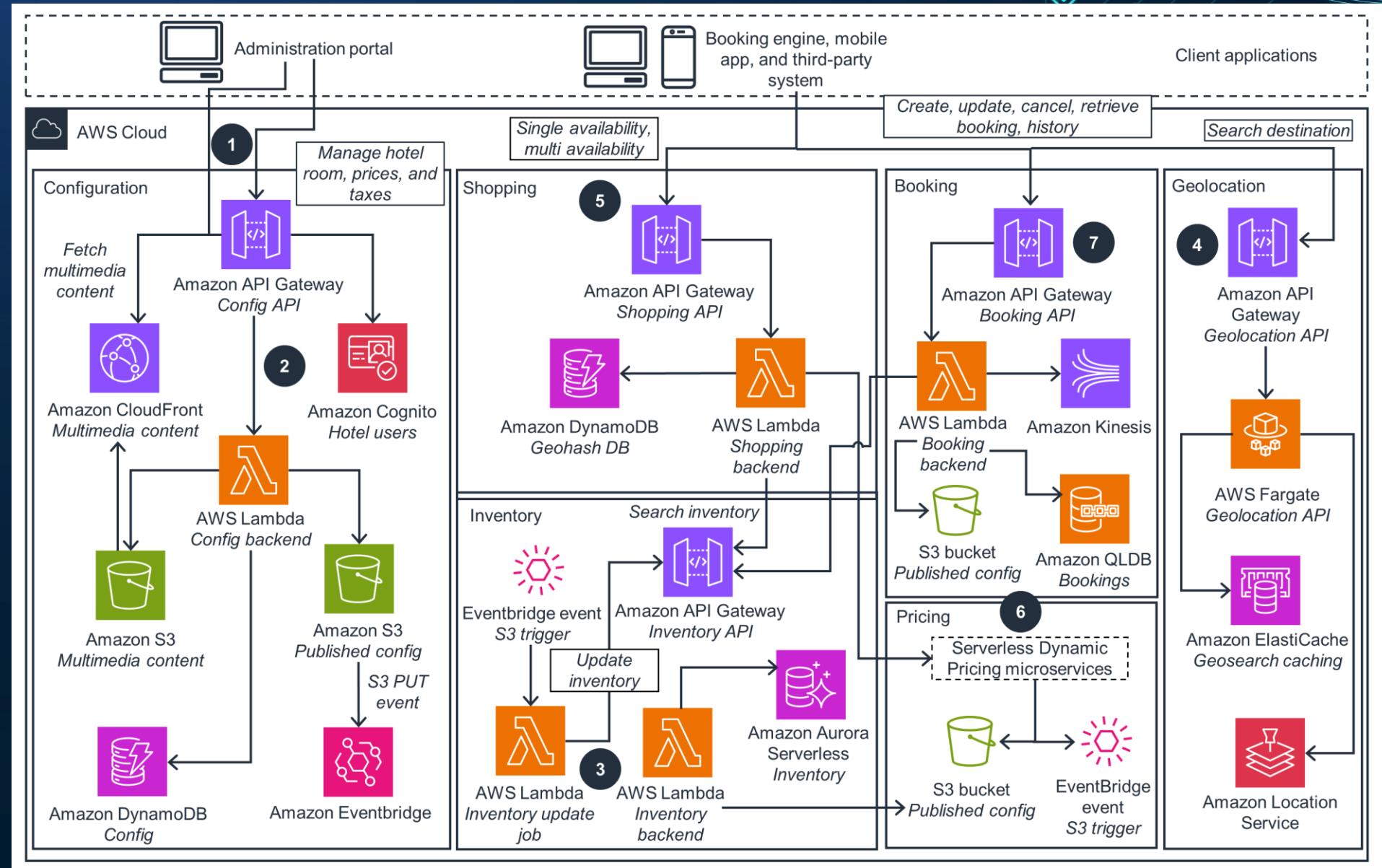




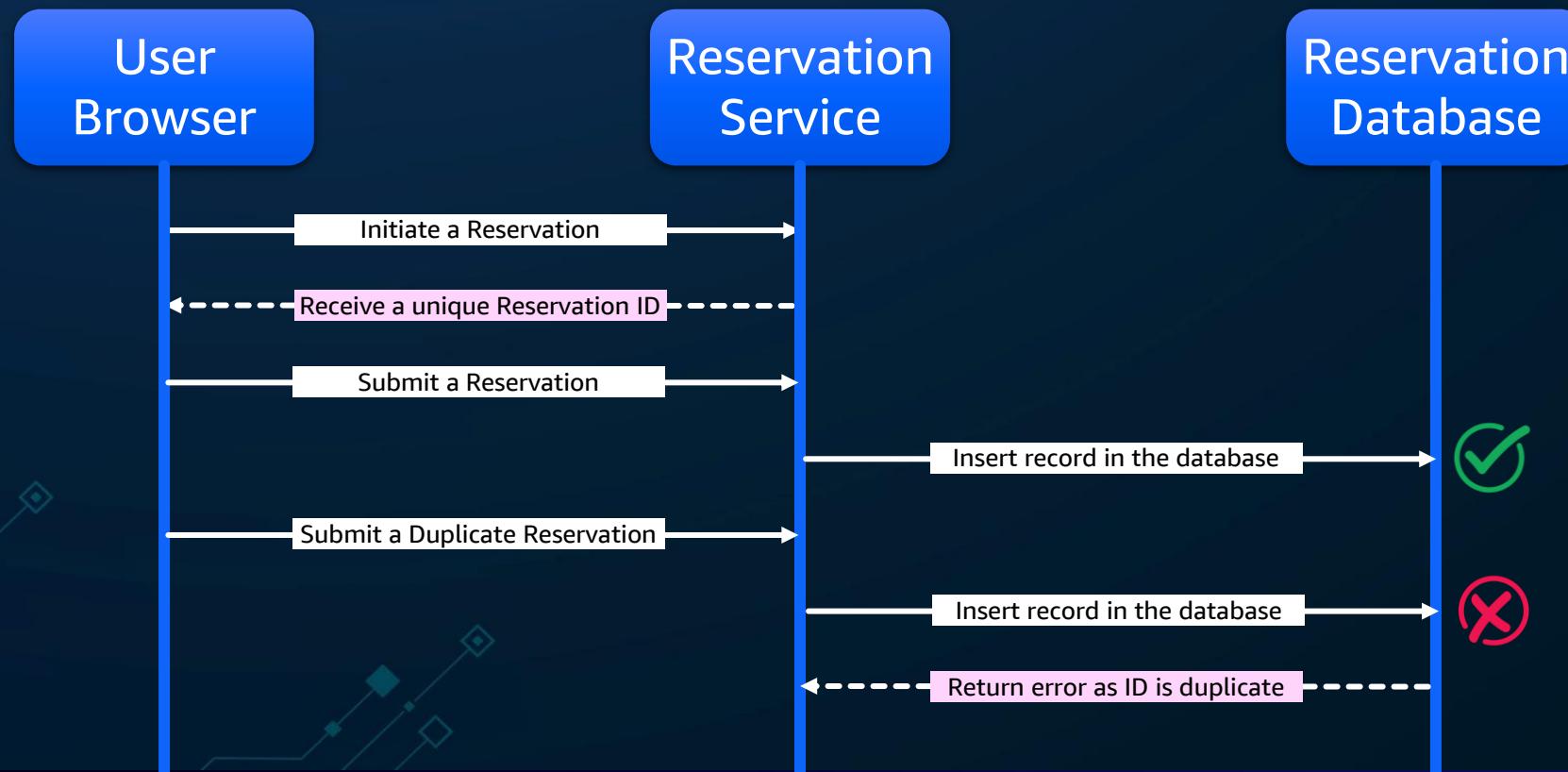
Photo Credit- <https://www.wsj.com/articles/the-mystery-of-vanishing-hotel-reservations-1436376090>

Double booking

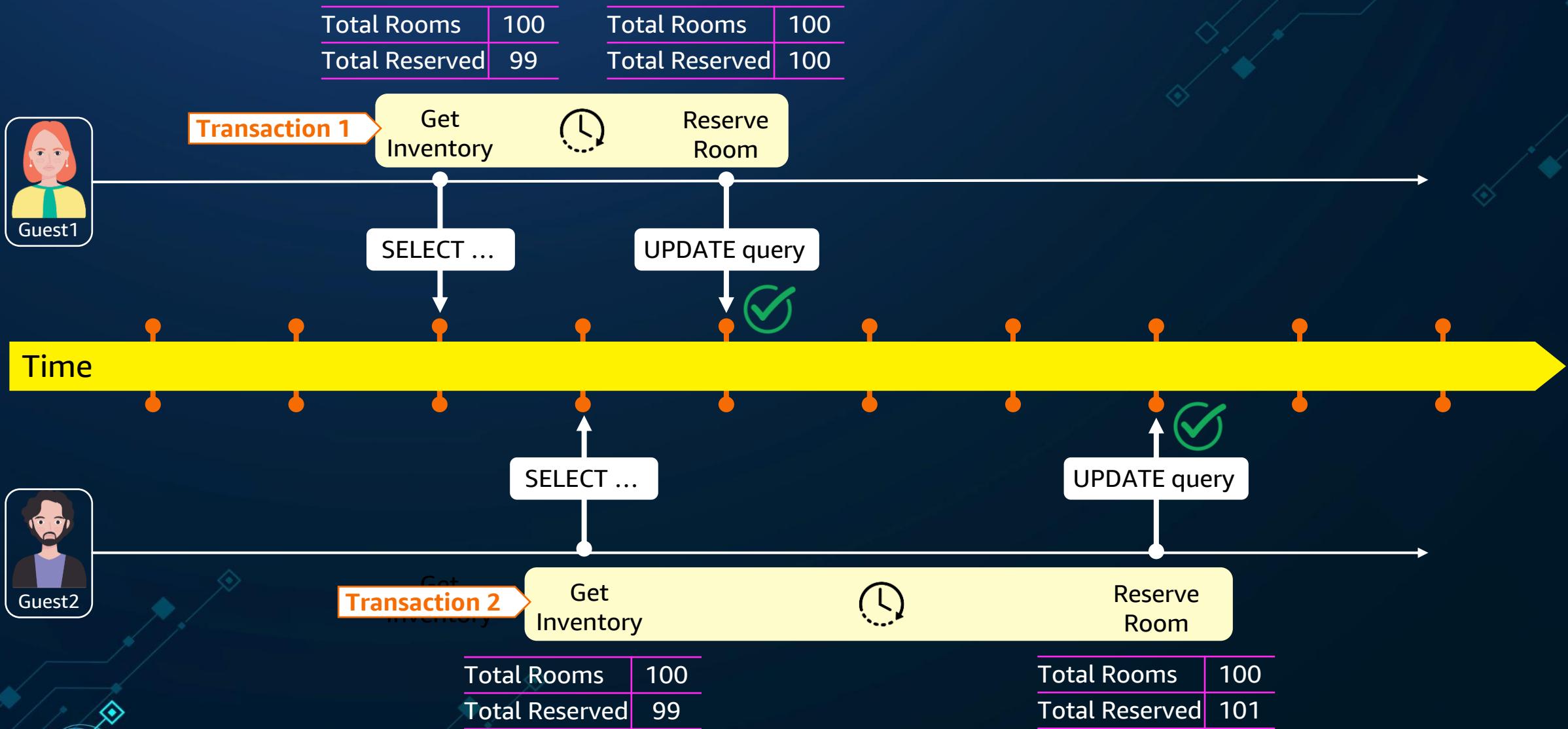
- Double-bookings occur when two bookings are made for the same slot, room, or table reservation.
- What causes double booking?
 - Listing the same property on multiple websites
 - Having multiple listing for the same property
 - Allowing late check-out or early check-in
 - User clicking on book button twice (multiple times)
 - Multiple user try to book the same property for the same dates

How to avoid double booking?

- User clicking on book button twice (multiple times)
 - Client Side implementation
 - Disable Reserve button after click
 - Idempotent Reservation ID



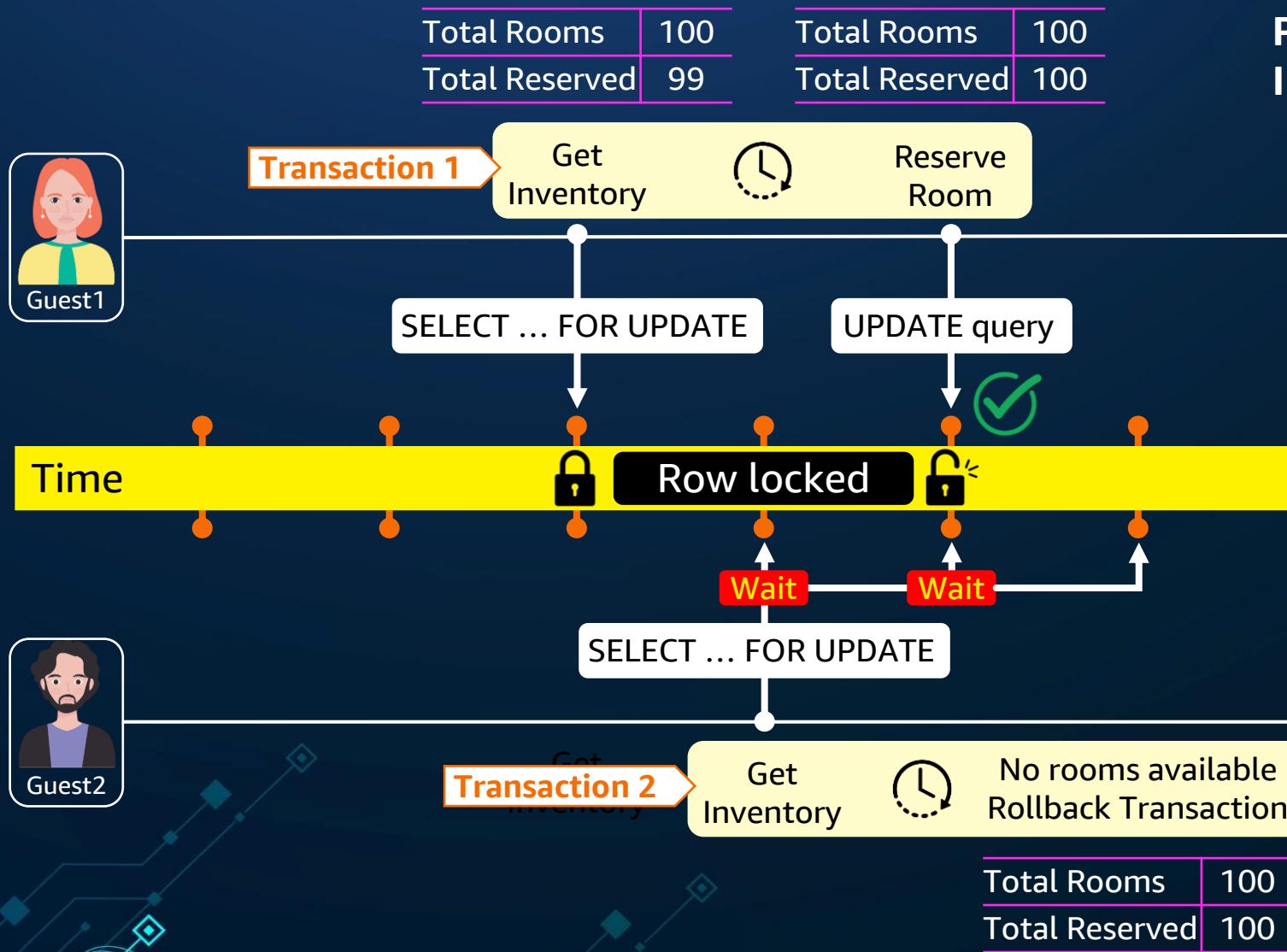
Double booking



How to avoid double booking?

- Multiple user try to book the same property for the same dates
 - Database locking
 - A mechanism that controls concurrent access to database resources to maintain data integrity and consistency. It restricts access to specific data elements (rows, tables, or objects) by one transaction while other transactions are waiting.
 - Pessimistic locking
 - Optimistic locking
 - Database constraints

Pessimistic locking



Please reserve this room for me now —
I'm still filling out my info."

Optimistic locking



Quick comparison

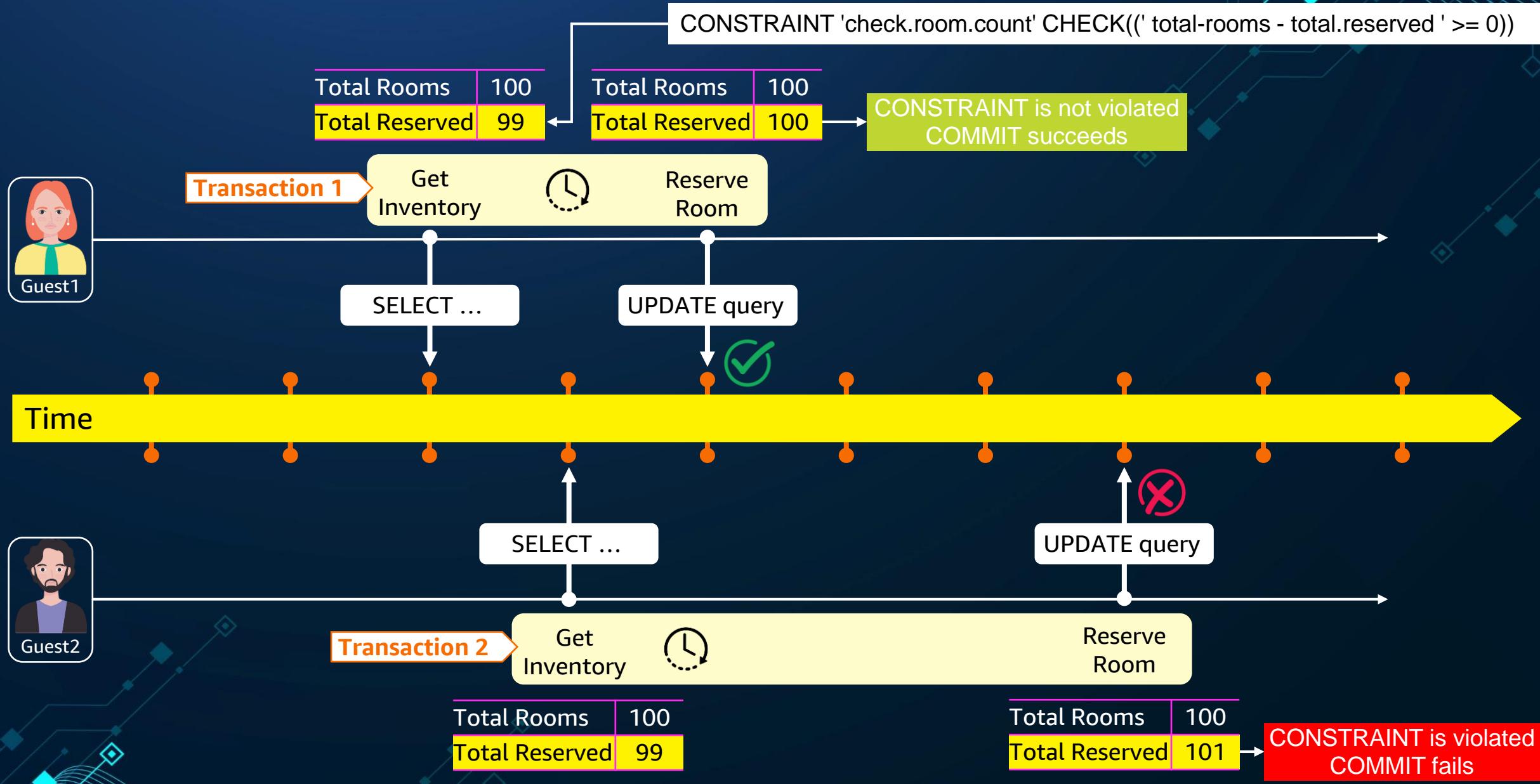


Aspect	Pessimistic Locking	Optimistic Locking
Basic Idea	Lock the resource immediately to prevent conflicts	Assume conflicts are rare, check before commit
Lock Timing	Lock acquired at start of transaction	Lock (or version check) done at commit time
Conflict Handling	Prevent conflicts by blocking others upfront	Detect conflicts after they happen and retry or abort
Performance	Can cause delays and reduce concurrency due to locks	Better concurrency, fewer delays if conflicts are rare
Complexity	Easier to implement	Requires conflict detection and retry logic
Typical Techniques	SELECT ... FOR UPDATE, exclusive locks	Version numbers, timestamps, compare-and-swap
Risk	Locks held too long can cause contention and blocking	Transactions may fail and need retry if conflict detected
Best For	High contention environments where conflicts are frequent	Low contention environments with rare conflicts
Example Use Cases	Bank account transfers, inventory systems with high write contention	Airline or hotel booking where conflicts are rare but possible
Suitability	When data integrity is critical and blocking is acceptable	When system performance and scalability are more important

How to avoid double booking?

- Multiple user try to book the same property for the same dates
 - Database locking
 - A mechanism that controls concurrent access to database resources to maintain data integrity and consistency. It restricts access to specific data elements (rows, tables, or objects) by one transaction while other transactions are waiting.
 - Pessimistic locking
 - Optimistic locking
 - Database constraints
 - A database constraint is a rule that restricts the type of data that can be stored in a table or column within a database.
 - They help maintain data integrity, consistency, and accuracy by enforcing certain conditions on the data, such as ensuring values are unique, not null, or within a specific range.

Database constraints





NEXT
STEPS

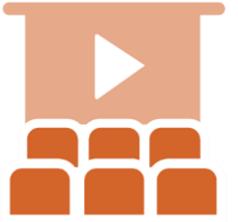


Similar systems



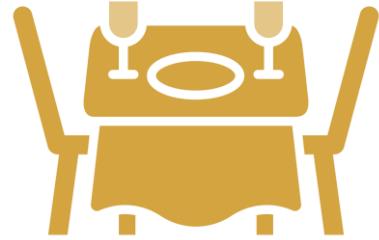
Travel & Transportation

- Hotel booking
- Flight reservation
- Parking lot reservation
- Car rentals



Entertainment & Events

- Movie ticket reservation
- Concerts and event ticketing
- Tour or activity booking



Food & Hospitality

- Restaurant reservations
- Festival or fair entry passes
- Lounge table reservations



Health & Wellness

- Doctor or clinic appointments
- Salon or spa appointments
- Gym class bookings

Domain	System	Listing Service Changes	Search Service Changes	Reservation Service Changes	Other Considerations
 Travel & Transportation	Flight Reservation	Flights, seats, fare classes, segments	Search by route, class, airline, fare	PNR creation, seat hold, pricing rules	Integrate with GDS , show seat maps, multi-city journey
	Car Rentals	Vehicles, locations, models, insurance	Search by pickup/drop, dates, type	Time-window reservations, deposit, license validation	Add driver ID, insurance payment, damage policy
	Parking Lot	Parking slot types, details (indoor, outdoor etc.)	Nearby location, date/time filter	Reserve hourly, enforce entry/exit windows	Sync with gate system, license plate entry
 Entertainment & Events	Movie Tickets	Movies, theatres, screens, seats	Search by movie, date, time, location	Lock seats for short time, cancel rules vary	Seat map UI, integrate with POS for concessions
	Concert/Event Tickets	Events, performers, venues, seat tiers	Filter by event, genre, venue	High concurrency, assign tiered seats	Sync with ticketing platform, QR code generation
	Tours/Activities	Tours, guides, capacity, equipment	Search by type, guide, group size	Group slots, equipment dependency	Weather sync (optional), waiver forms

Domain	System	Listing Service Changes	Search Service Changes	Reservation Service Changes	Other Considerations
 Food & Hospitality	Restaurant Booking	Tables, seating plans, shifts	Search by time, capacity, cuisine	Reserve table with time slot, buffer handling	Table layout UI, add waitlist support
	Festival/Fair Passes	Events, sessions, access zones, ticket tiers	Search by location, date, performer /event line-up	Reserve based on date/session/tier, gate access rules	QR codes, dynamic pricing, sync with scanners
	Lounge Table Reservations	Table types, VIP status, service levels, floor maps	Search by time, group size, location, table type	Time-slot booking, min spend, merge tables if needed	Floor plan UI, min spend enforcement, optional pre-orders
 Health & Wellness	Doctor Appointments	Providers, visit types, availability	Search by specialty, insurance, location	Slot locking, patient match, pre-visit forms	EMR sync via FHIR/HL7, patient data encryption
	Salon/Spa	Services, staff, duration	Search by stylist, type, timing	Bundle services (e.g., cut + color), staff mapping	Add resource locking (chairs/tools), ratings
	Gym/Sports Booking	Classes, trainers, facility schedule	Search by type, capacity, time	Register for classes, recurring booking	Add membership validation, access badge link