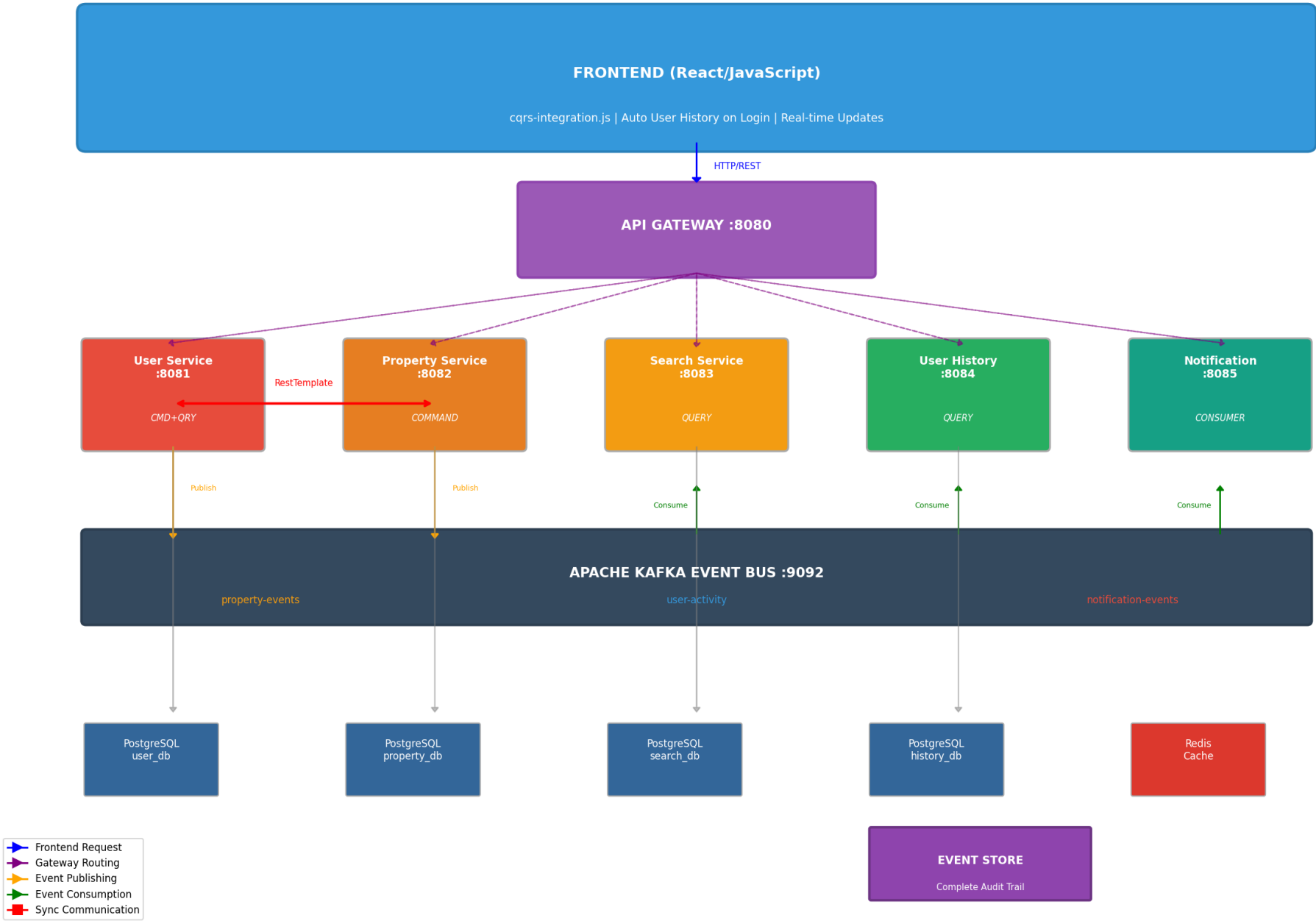


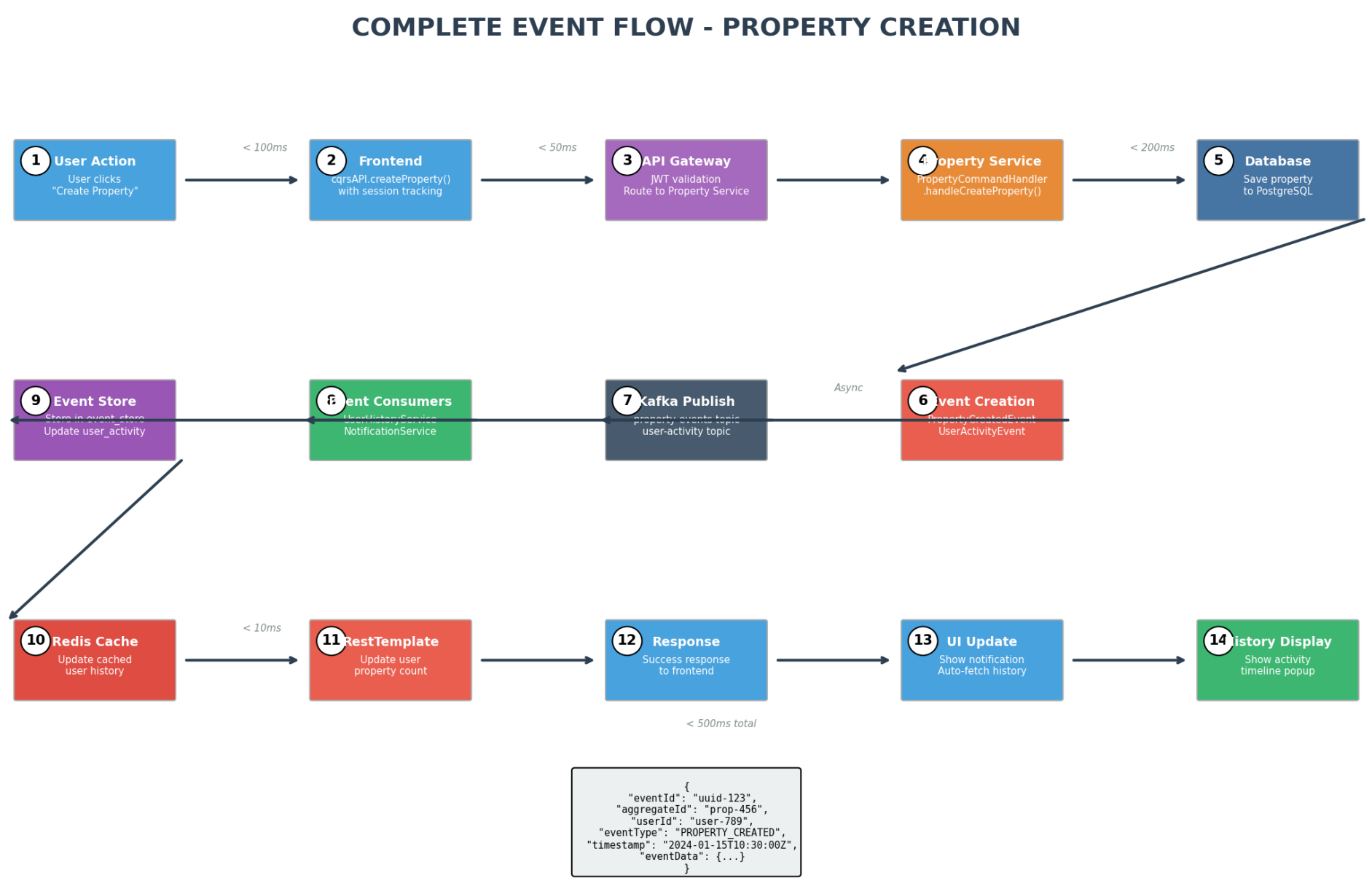
1. Complete System Architecture

HYDERABAD INFRA - COMPLETE CQRS ARCHITECTURE

Event-Driven Microservices with Complete Flow



2. Complete Event Flow - Property Creation



3. API Endpoints & Data Flow

COMPLETE API ENDPOINTS & DATA FLOW

User Service :8081		
POST	/api/auth/register	(User Registration)
POST	/api/auth/login	(User Login + History Fetch)
GET	/api/users/{id}	(Get User Profile)
PUT	/api/users/{id}	(Update User)
POST	/api/users/{id}/property-count	(Increment Property Count)

Property Service :8082		
POST	/api/properties	(Create Property (CQRS Command))
GET	/api/properties/{id}	(View Property (Tracked))
PUT	/api/properties/{id}	(Update Property)
DELETE	/api/properties/{id}	(Delete Property)
GET	/api/properties/user/{userId}	(User Properties)

Search Service :8083		
GET	/api/search/properties	(Search Properties)
POST	/api/search/advanced	(Advanced Search)
GET	/api/search/filters	(Get Search Filters)
GET	/api/search/suggestions	(Auto-suggestions)
GET	/api/search/history/{userId}	(Search History)

User History :8084		
GET	/api/user-history/{userId}	(Complete History)
GET	/api/user-history/{userId}/recent	(Recent Activities)
GET	/api/user-history/{userId}/properties	(Property Activities)
GET	/api/user-history/{userId}/searches	(Search History)
GET	/api/user-history/{userId}/events	(Raw Events)

Notification :8085		
POST	/api/notifications/email	(Send Email)
POST	/api/notifications/sms	(Send SMS)
GET	/api/notifications/{userId}	(User Notifications)
PUT	/api/notifications/{id}/read	(Mark as Read)
GET	/api/notifications/preferences	(Get Preferences)

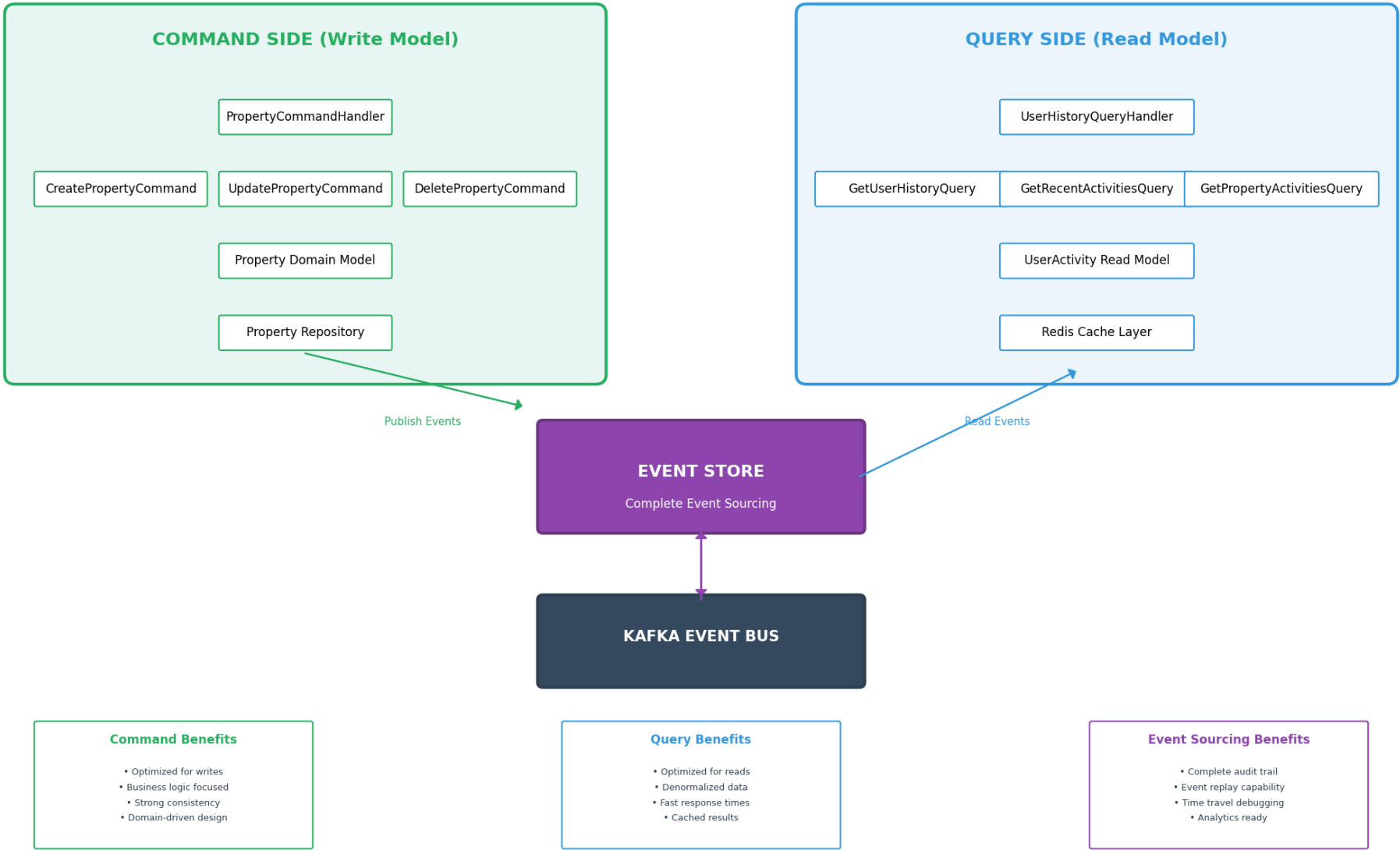
File Upload :8086		
POST	/api/files/upload	(Upload File)
GET	/api/files/{id}	(Download File)
DELETE	/api/files/{id}	(Delete File)
POST	/api/files/bulk	(Bulk Upload)
GET	/api/files/property/{id}	(Property Files)

Data Flow Patterns:

Command Flow:	Frontend → Gateway → Service → Database → Event → Kafka
Query Flow:	Frontend → Gateway → Service → Cache/Database → Response
Event Flow:	Service → Kafka → Consumers → Event Store → Read Models
Sync Flow:	Service → RestTemplate → Target Service → Response

4. CQRS Pattern Implementation

CQRS PATTERN IMPLEMENTATION



5. User Journey - Login to History Display

USER JOURNEY - LOGIN TO HISTORY DISPLAY

Complete flow showing how users get their history on login



Total Time: < 2 seconds from login to history display

Key Components Involved:

Frontend (cqrs-integration.js) → API Gateway → User Service → User History Service → Redis/PostgreSQL → Event Store