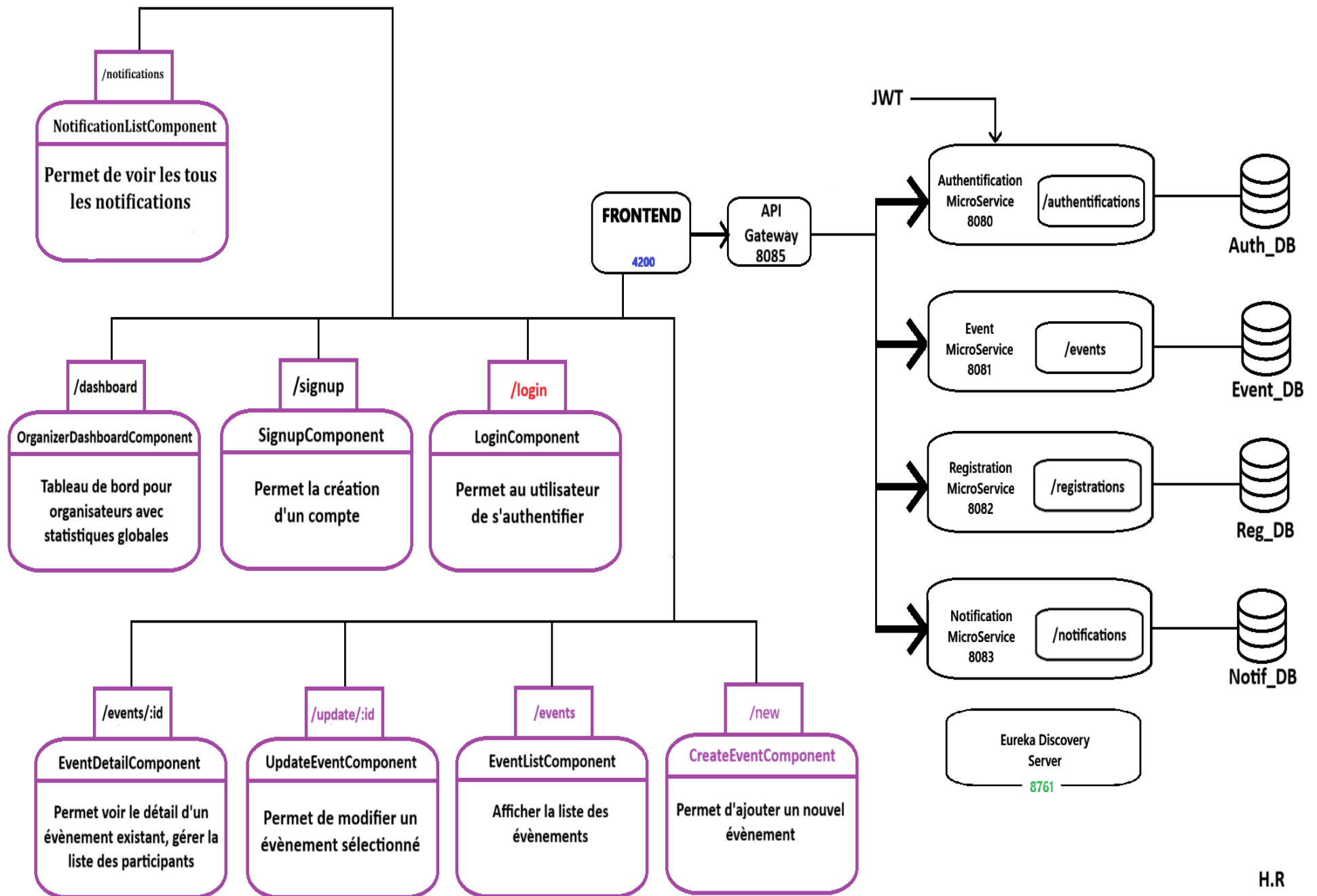


# Architecture Complete





## Les MicroServices

MicroService	Description	Entité(s)	Endpoints Rest
<b>Authentication MicroService 8080</b>	Gestion des Authentications	<b>User</b> { id, username, email, password, <b>Role</b> } <b>Role</b> { Organizer, Participant }	<b>POST</b> /authentications/signin <b>POST</b> /authentications/signup <b>GET</b> /authentications/user <b>GET</b> /Internal/users/batch
<b>Event MicroService 8081</b>	Gestion des Événements	<b>Event</b> { id, title, eventDate, eventTime, location, <b>EventType</b> , organizerId, views, createdAt, maxCapacity, <b>EventStatus</b> , <b>EventMedia</b> } <b>EventType</b> { Concert, Formation, JournéeScientifique, JournéeCulturelle, Conference } <b>EventMedia</b> { id, eventId, filepath, mediatype, title } <b>EventStatus</b> { Active, Ended, Canceled }	<b>POST</b> /events <b>PUT</b> /events/{id} <b>Delete</b> /events/{id} <b>POST</b> /events/{id}/media <b>GET</b> /events/{id}/stats <b>GET</b> /events <b>GET</b> /events/{id} <b>GET</b> /events/media/{fileName} <b>GET</b> /Internal/upcoming <b>GET</b> /Internal/{id}/organizer <b>Delete</b> /events/{eventide}/media
<b>Registration MicroService 8082</b>	Gestion des Participations	<b>Registration</b> { id, userid, eventId, registeredAt, <b>RegistrationStatus</b> } <b>RegistrationStatus</b> { Registered, Canceled, Refused, Enattend }	<b>POST</b> /registrations/{eventId} <b>DELETE</b> /registrations/{eventId} <b>GET</b> /registrations/{eventide}/participantsCount <b>GET</b> /registrations/{eventId}/status <b>GET</b> /registrations/{eventId}/registered <b>POST</b> /registrations/{eventId}/accept/{userId} <b>POST</b> /registrations /{eventId}/refuse/{userId} <b>GET</b> /registrations/{eventId}/pending <b>GET</b> /registrations/{eventId}/accepted <b>GET</b> /registrations/{eventId}/refused <b>GET</b> /Internal/{eventId}/registeredUsers

<b>Notification MicroService 8083</b>	Gestion des Notifications	<b>Notification</b> { id, userid, eventid, title, message, type, createdAt, isread } <b>ReminderTracking</b> { id, userid, eventid, sent }	<b>GET</b> /notifications <b>PATCH</b> /notifications/{id} <b>POST</b> /Internal/triggerRegistration <b>POST</b> /Internal/triggerRegistrationDecision <b>POST</b> /Internal/triggerUnregistration <b>POST</b> /Internal/triggerEventUpdate <b>POST</b> /Internal/triggerEventCancel
---	------------------------------	--	--





## Les Composants web

Composant	Description	Routes
CreateEventComponent	Formulaire de création d'un nouvel évènement	/new
UpdateEventComponent	Modification d'un évènement existant	/update/:id
EventListComponent	Affichage de la liste des évènements, Recherche des évènements, Bouton de modification des évènements.	/events
EventDetailComponent	Affichage du détail d'un évènement, Boutons de modification/ Annulation d'un évènement, Boutons de demande d'inscription/Désinscription a un évènement, Afficher la liste des Participants Acceptés/ En attend /Refusé, Boutons pour accepter/Refuser une demande d'inscription	/events/:id
OrganizerDashboardComponent	Tableau de bord pour organisateurs avec statistiques globales, avec gestion des participants	/dashboard
LoginComponent	Authentification	/login
SignupComponent	Creation d'un compte	/signup
NotificationListComponent	Voir tous les notifications	/notifications
AppComponent	Conteneur principal avec <b>router-outlet</b>	{racine}
NotificationComponent	Composant global permettant l'affichage des notifications en temps réel pour l'utilisateur (inscription acceptée, mise à jour d'un évènement, rappel, etc.). Il est intégré dans la racine pour être visible dans l'ensemble de l'application.	Aucune Route

<b>Confirm-dialog</b>	Un globale permettant de afficher un dialog de confirmation pour une fonctionnalité, il est aussi Il est intégré dans la racine pour être visible dans l'ensemble de l'application.	<b>Aucune Route</b>
-----------------------	---	---------------------

**FIN**