Authentication Flow:

Users authenticate through Azure Active Directory (Azure AD) using OpenID Connect (OIDC) for identity federation.

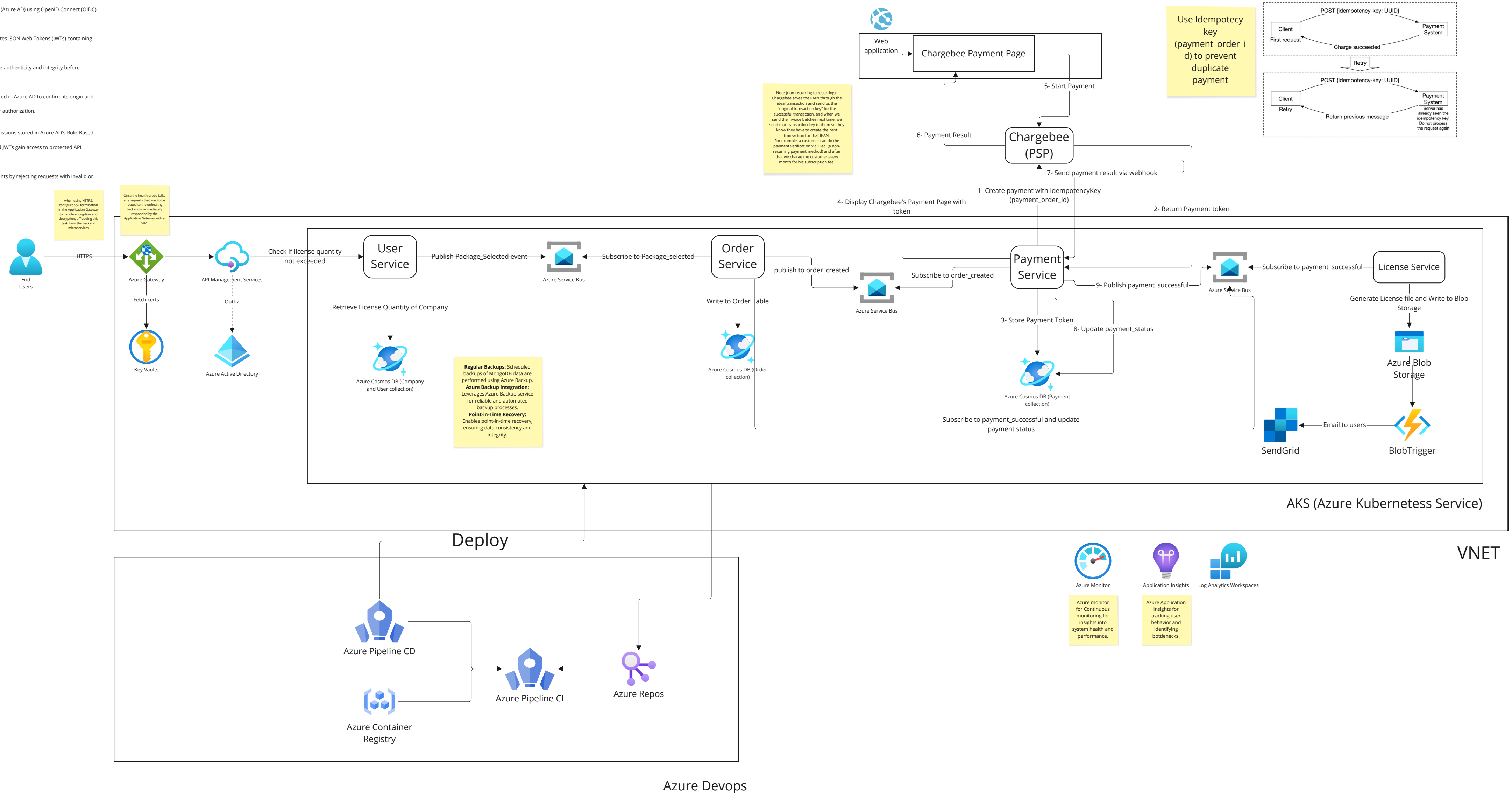
JWT Usage:
After successful authentication, the system generates JSON Web Tokens (JWTs) containing user claims and roles.

API Gateway Integration:
The API Gateway validates incoming JWTs to ensure authenticity and integrity before processing requests.

Token Validation Process:
Validates the JWT's signature using public keys stored in Azure AD to confirm its origin and prevent tampering.
Checks token expiration to ensure it is still valid for authorization.

Authorization Decision:
Authorizes requests based on user roles and permissions stored in Azure AD's Role-Based Access Control (RBAC).
Only authenticated and authorized users with valid JWTs gain access to protected API endpoints.

Secure Communication:
Ensures secure communication between components by rejecting requests with invalid or expired tokens.


Web application
Chargebee Payment Page
5- Start Payment
6- Payment Result
Chargebee (PSP)
7- Send payment result via webhook
1- Create payment with IdempotencyKey (payment_order_id)
4- Display Chargebee's Payment Page with token
2- Return Payment token
End Users
HTTPS
Azure Gateway
Fetch certs
Key Vaults
API Management Services
Outh2
Azure Active Directory
Check If license quantity not exceeded
User Service
Publish Package_Selected event
Azure Service Bus
Subscribe to Package_selected
Order Service
publish to order_created
Azure Service Bus
Subscribe to order_created
Payment Service
9- Publish payment_successful
Azure Service Bus
Subscribe to payment_successful
License Service
Retrieve License Quantity of Company
Azure Cosmos DB (Company and User collection)
Write to Order Table
Azure Cosmos DB (Order collection)
3- Store Payment Token
8- Update payment_status
Azure Cosmos DB (Payment collection)
Generate License file and Write to Blob Storage
Azure Blob Storage
Subscribe to payment_successful and update payment status
SendGrid
Email to users
BlobTrigger
AKS (Azure Kubernetess Service)
VNET
Deploy
Azure Pipeline CD
Azure Container Registry
Azure Pipeline CI
Azure Repos
Azure Devops
Azure Monitor
Application Insights
Log Analytics Workspaces
POST {idempotency-key: UUID}
Client
First request
Payment System
Charge succeeded
Retry
POST {idempotency-key: UUID}
Client
Retry
Payment System
Server has already seen the idempotency key. Do not process the request again
Return previous message

when using HTTPS, configure SSL termination in the Application Gateway to handle encryption and decryption, offloading this task from the backend microservices

Once the health probe fails, any requests that was to be routed to the unhealthy backend is immediately responded by the Application Gateway with a 502.

Note (non-recurring to recurring): Chargebee saves the IBAN through the ideal transaction and send us the "original transaction key" for the successful transaction. and when we send the invoice batches next time, we send that transaction key to them so they know they have to create the next transaction for that IBAN.
For example, a customer can do the payment verification via iDeal (a non-recurring payment method) and after that we charge the customer every month for his subscription fee.

Use Idempotecy key (payment_order_id) to prevent duplicate payment

Regular Backups: Scheduled backups of MongoDB data are performed using Azure Backup.
Azure Backup Integration: Leverages Azure Backup service for reliable and automated backup processes.
Point-in-Time Recovery: Enables point-in-time recovery, ensuring data consistency and integrity.

Azure monitor for Continuous monitoring for insights into system health and performance.

Azure Application Insights for tracking user behavior and identifying bottlenecks.