

Follow for

Daily Tech tips

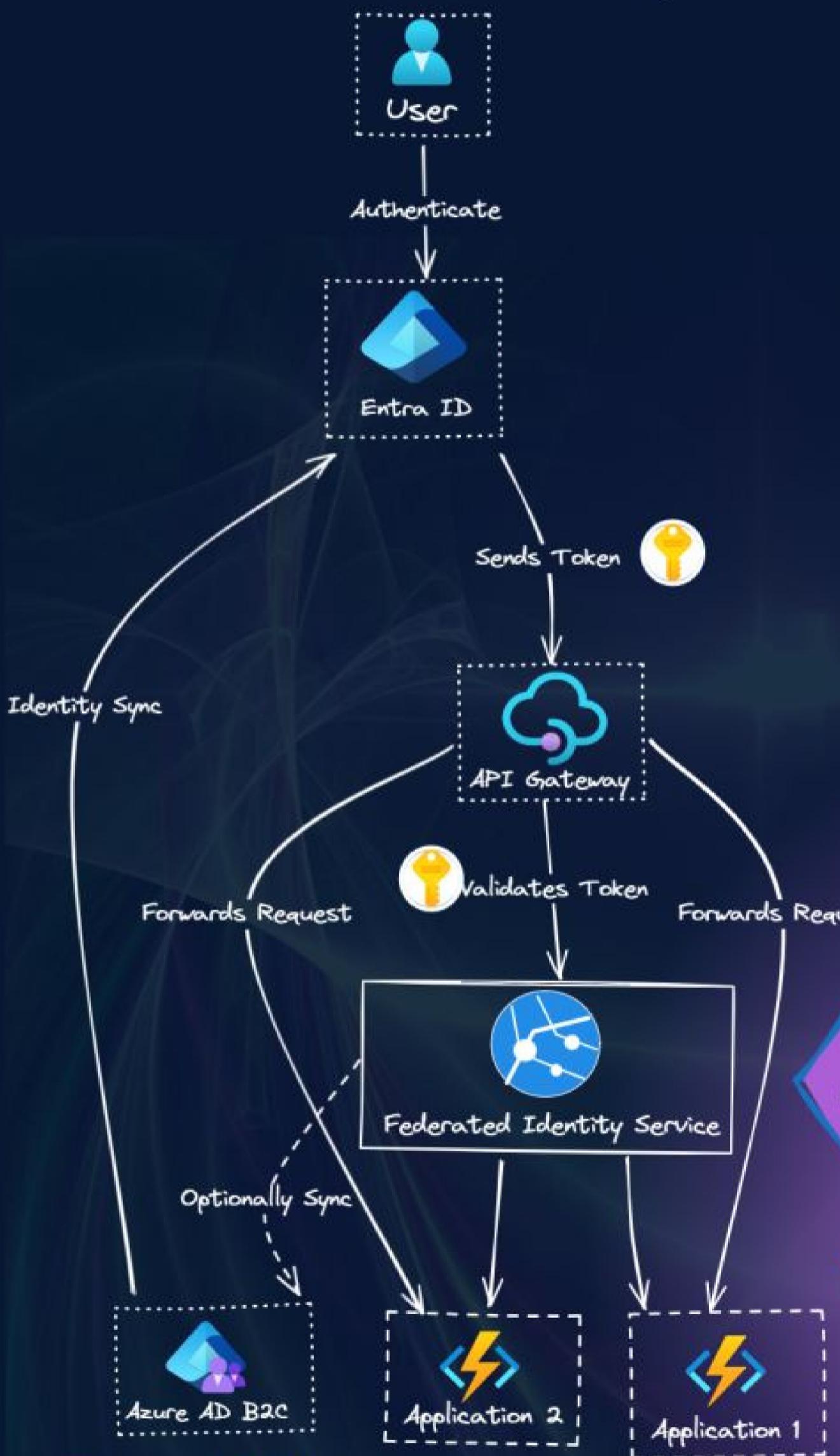


Repost

16 CLOUD DESIGN PATTERNS



Federated Identity pattern



- ✓ The user logs in just once using an identity provider.
- ✓ Azure's identity management service steps in, handling user identities and permissions.
- ✓ API Gateway checks that the user's token is valid, so no one gets in who shouldn't.
- ✓ Federated Identity Service is a central hub that makes sure the user's identity can be used across different apps.
- ✓ Whether it's App 1, 2, or 3, they all know the user is who they say they are, thanks to the identity service.

Anti-corruption Layer pattern



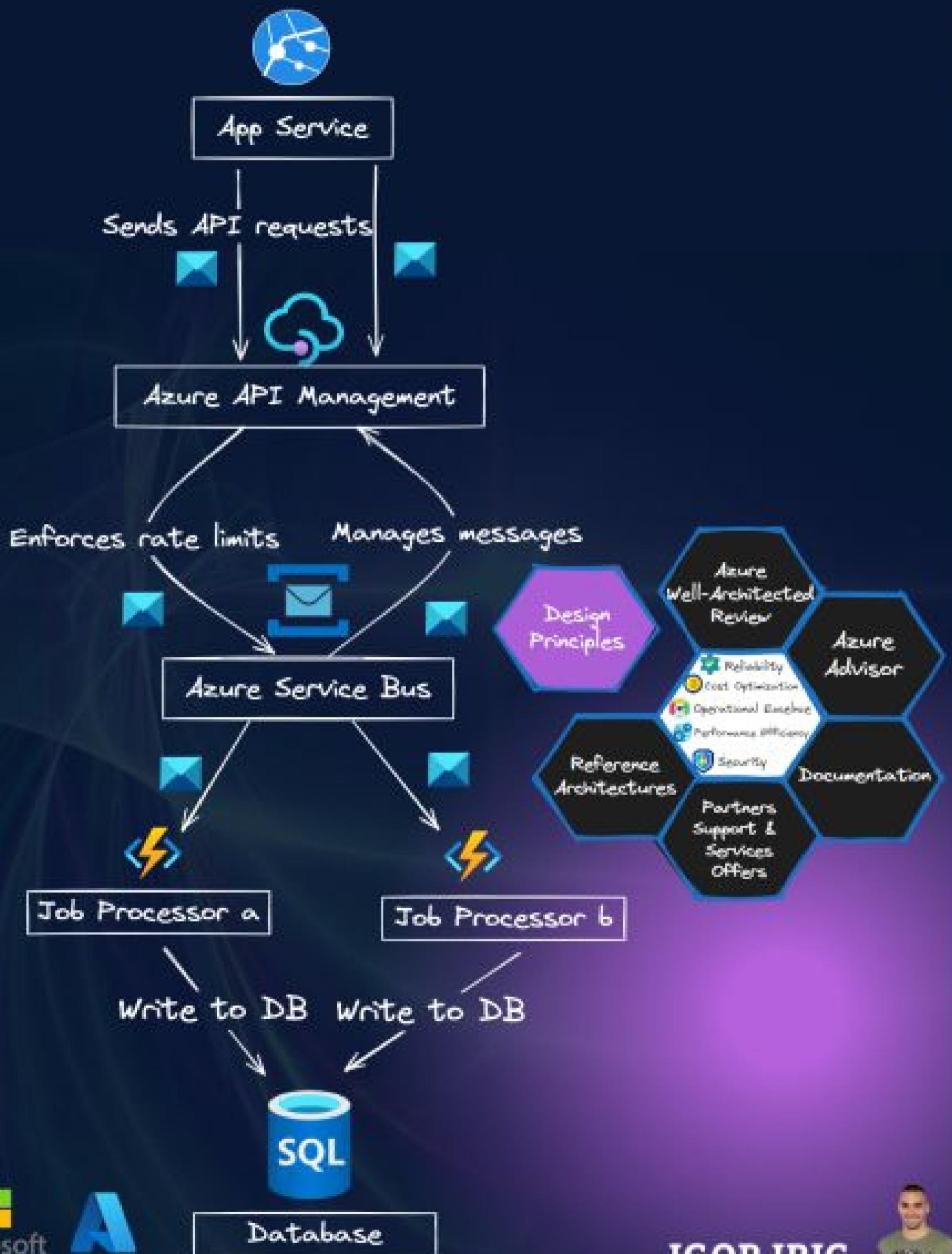
- ✓ Azure API Management takes requests from clients and directs them where they need to go.
- ✓ Azure Functions takes the new requests and converts them into a language the old system understands and also processes messages from and to Azure Service Bus.
- ✓ Azure Service Bus carries the translated requests to the new system and makes sure the responses get back.
- ✓ Cosmos DB is for new data, and Azure SQL holds the old data.

Quarantine pattern



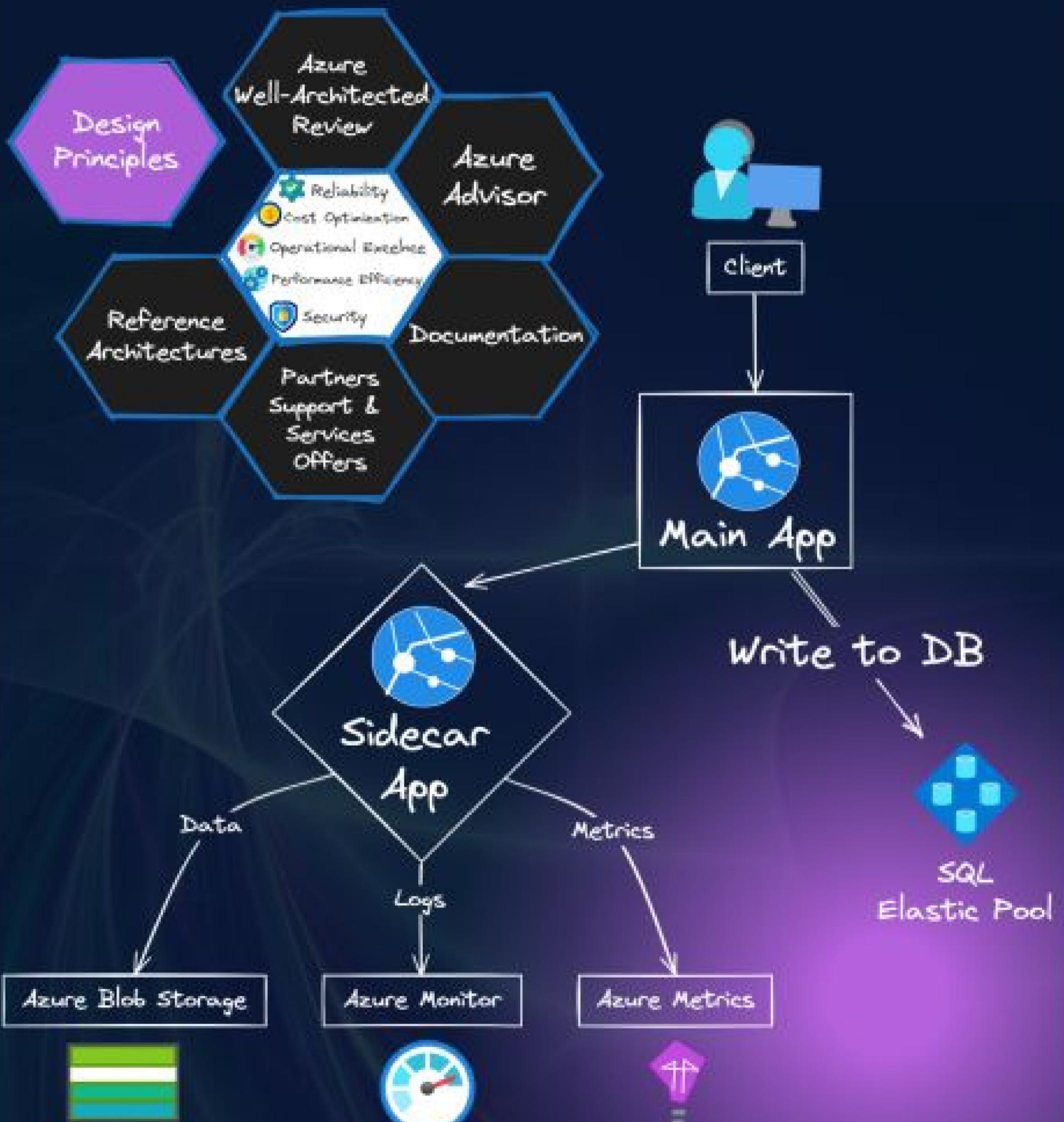
- ✓ Clients send data to your App Service, acting as your front end.
- ✓ Azure Logic Apps, your Validation Service, checks if the data is good or bad.
- ✓ Good data passes forward to Azure Function, your Processing Service, doing the transformation and saving of the data.
- ✓ Once the data is processed, it rests in Azure Cosmos DB, your Data Store.
- ✓ The bad data? It's sent to Azure Blob Storage, the Quarantine Service, keeping your environment stable and healthy.
- ✓ The quarantined data isn't discarded or deleted it's reviewed and resolved through the Azure Portal, giving it a second chance.
- ✓ Corrected data re-enters the flow, back to the Processing Service, to fulfill its purpose.

Rate Limiting pattern



- ✓ App Service is the starting point, handling your business operations.
- ✓ Azure API Management steps in, putting caps on API requests to prevent traffic jams.
- ✓ Azure Service Bus orchestrates the message traffic, keeping everything in check.
- ✓ Job Processors, a & b, stay busy processing tasks, making sure work gets done.
- ✓ SQL Database stands strong, storing all the outcomes of your app's hard work.

Sidecar pattern



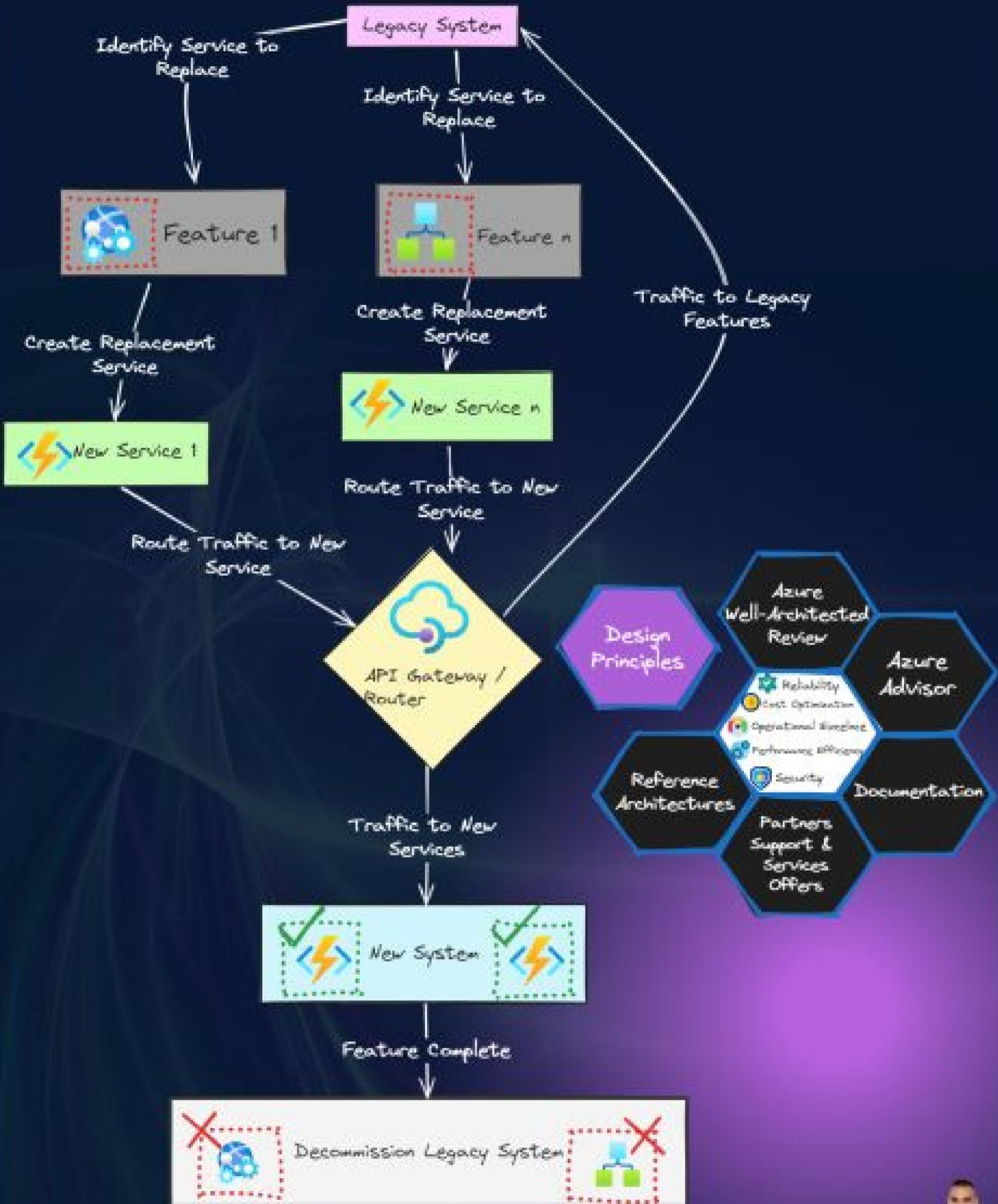
A

IGOR IRIC



- ✓ Main Application remains the core, focused on its business logic.
- ✓ Sidecar App attaches to the main app, to include logging, monitoring, and more.
- ✓ Azure Blob Storage integrates with the sidecar for data management.
- ✓ Azure Monitor integrates with the sidecar to log activities without breaking everything.
- ✓ Azure Metrics syncs for real-time performance and health checks.
- ✓ The main app continues uninterrupted, efficiently writing to an SQL Elastic Pool.

A Strangler Fig pattern



- ✓ Identify service to replace in the legacy system.
- ✓ Create replacement service that replicates the functionality of the legacy feature.
- ✓ Route traffic to new service instead of the legacy one using API Gateway or Router.
- ✓ Maintain routes to legacy features that have not yet been replaced.
- ✓ Use tools like Azure Advisor to ensure the new service is reliable, performant, and cost-effective.
- ✓ Once all features have been replaced, the new system should be complete and fully functional.
- ✓ After a period of parallel running and once confident in the new system's capabilities, decommission the legacy system.

CQRS cloud pattern



Design
Principles

Azure
Well-Architected
Review

Azure
Advisor

Reference
Architectures

Partners
Support &
Services
Offers

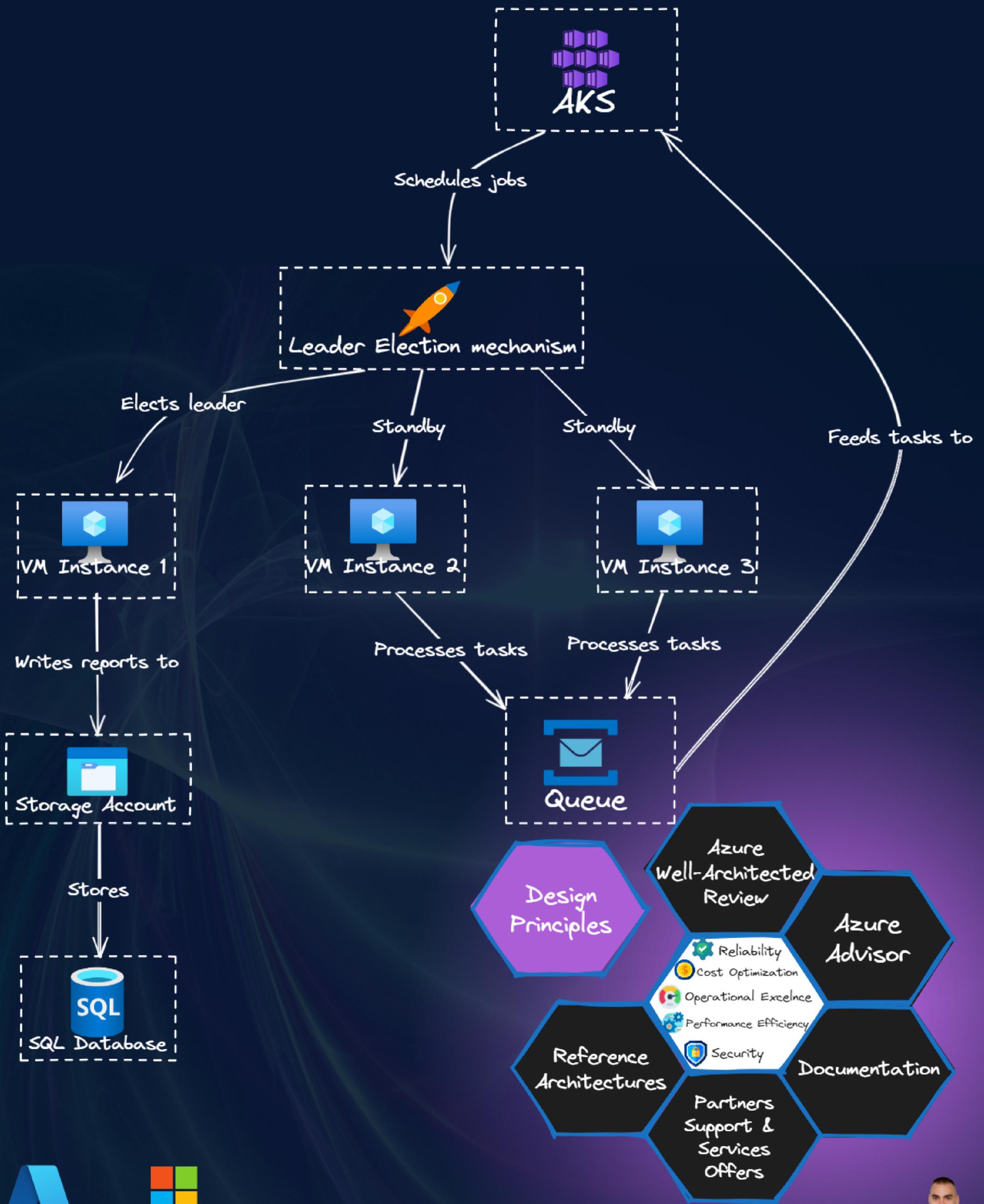
Documentation

- Reliability
- Cost Optimization
- Operational Excellence
- Performance Efficiency
- Security



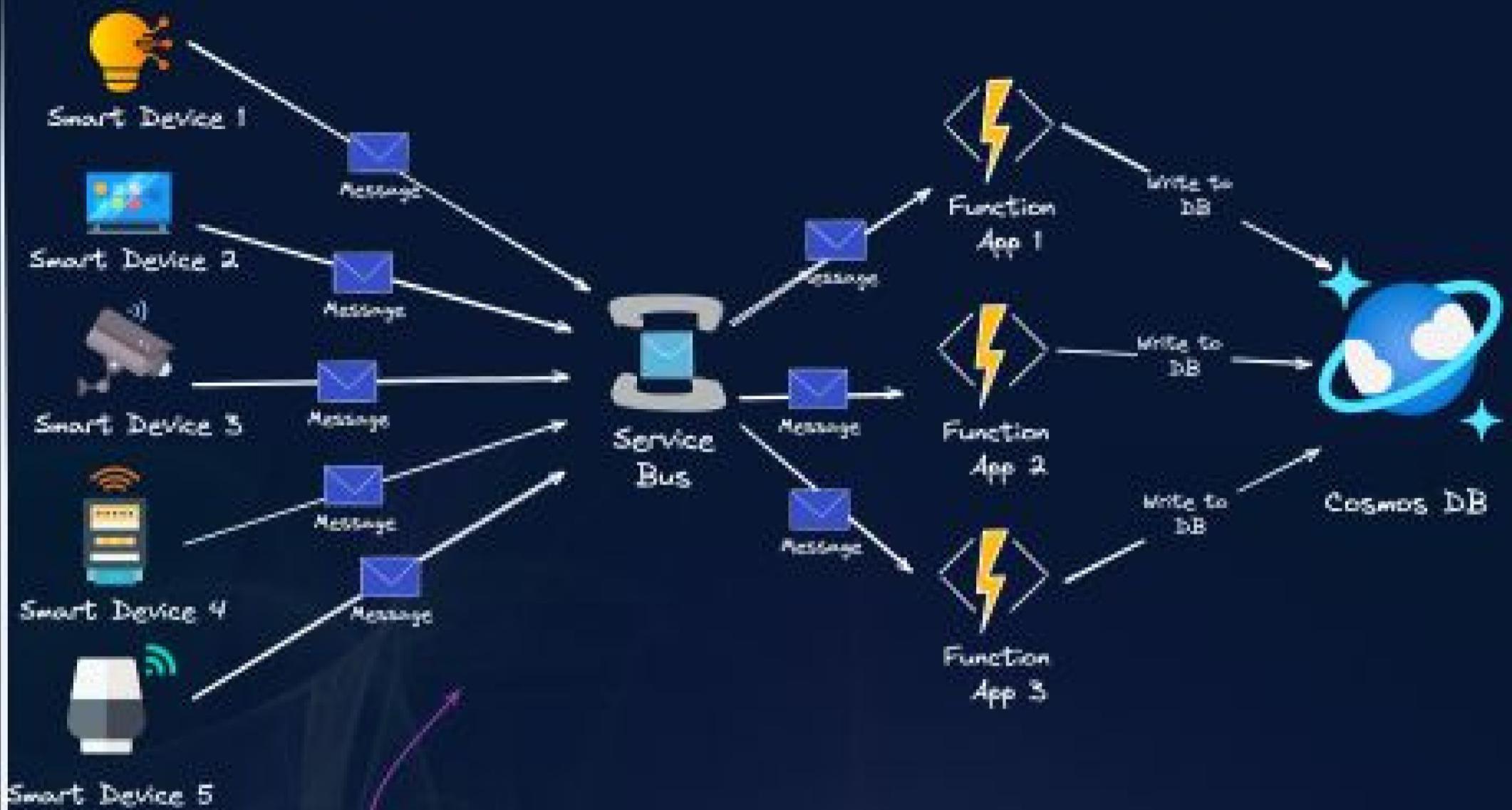
- ✓ Clients issue commands or queries, initiating interactions with your system.
- ✓ API Management acts as a gateway, intelligently routing commands and queries.
- ✓ Command Handlers are tasked with processing requests to change data, ensuring each action is captured with precision.
- ✓ Azure Event Hubs plays a crucial role, distributing information about state changes across the system.
- ✓ Command DB is the source of truth for writes, thoroughly tracking state mutations.
- ✓ Query Handlers focus mostly on retrieval, delivering data rapidly and efficiently.
- ✓ Query DB is optimized for reads, providing quick access to the latest information.
- ✓ Update Processors keeps the data landscape consistent, aligning the Command and Query sides.

Leader Election pattern



- ✓ Azure Kubernetes Service (AKS) is like a manager, setting up jobs for the work.
- ✓ The Leader Election mechanism is the audition process, picking out the VM to lead the process.
- ✓ One VM steps into the spotlight, taking the lead on doing all the important jobs it's elected to finish.
- ✓ The other VMs aren't out of the job – they're the supporting cast, tackling all the other tasks.
- ✓ The leading VM then puts the finishing touches by safely storing all the data in the storage room.
- ✓ While this is happening, the rest of the VMs keeps the work running smoothly, making sure everything's ready for the next job.

Competing Consumers cloud pattern



Design
Principles

Azure
Well-Architected
Review

Azure
Advisor

Reference
Architectures

Documentation

Partners
Support &
Services
Offers



5 smart IoT devices sending messages to Azure Service Bus.

- These messages are picked up by three Azure Functions.
- Each Function processes messages and sends metadata to Azure Cosmos DB.

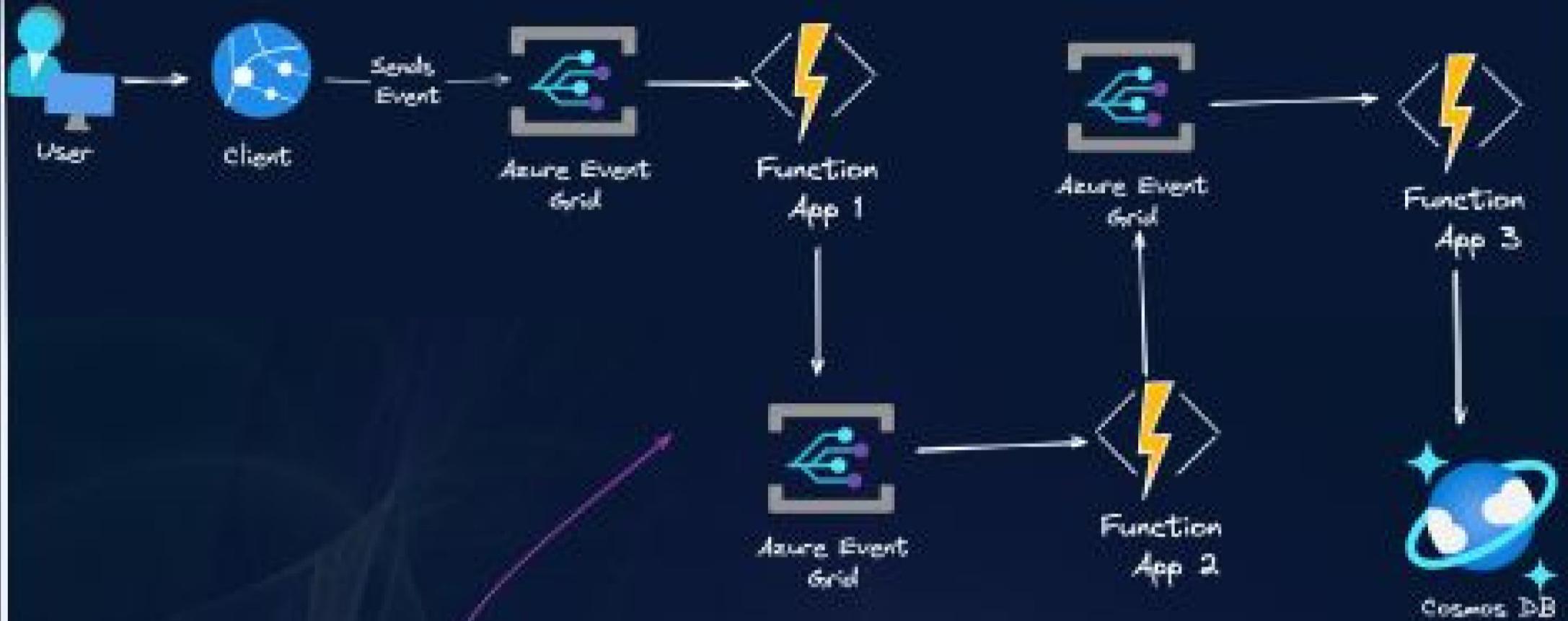
Now, why use the Competing Consumers pattern here?

 Efficient message processing: Multiple consumers handle messages concurrently, speeding up processing.

 Load balancing: The workload is distributed across consumers, preventing overload.

 Scalability: Easily scale the number of consumers based on the workload.

Choreography cloud pattern



Design Principles

Azure Well-Architected Review

Azure Advisor

Reference Architectures

Partners Support & Services Offers

Documentation

- Reliability
- Cost Optimization
- Operational Excellence
- Performance Efficiency



Search over my profile for
more explanations and hit
Like, Repost and comment.

Claim check cloud pattern



Design Principles

Azure Well-Architected Review

Azure Advisor

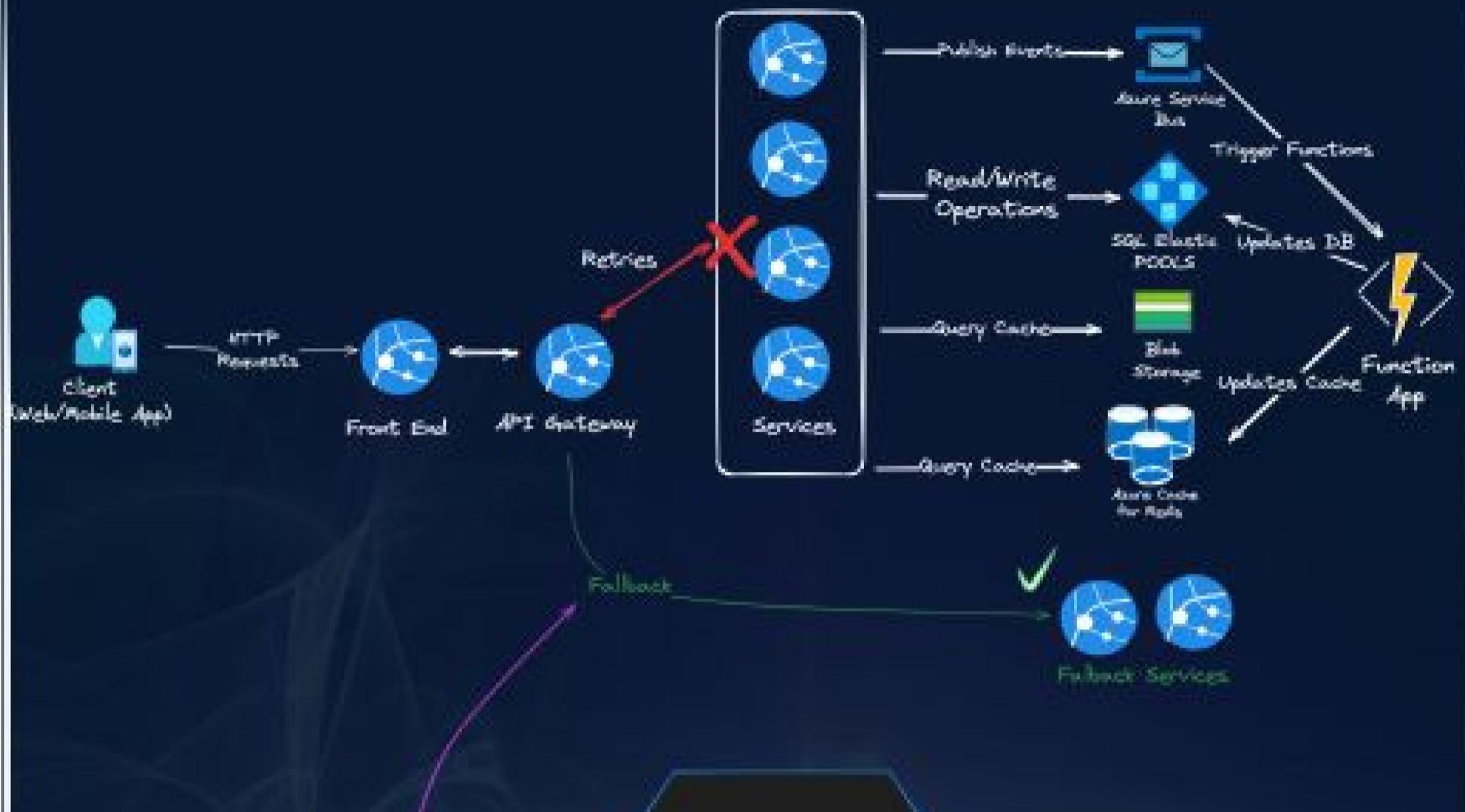
Reference Architectures

Partners Support & Services Offers

Documentation



Circuit breaker cloud pattern



Design Principles

Azure Well-Architected Review

Azure Advisor

Reference Architectures

Partners Support & Services Offers

Documentation



Bulkhead cloud pattern



Cache-Aside cloud pattern



Backends for Frontends pattern



Design Principles

Azure Well-Architected Review

Azure Advisor

Reference Architectures

Partners Support & Services Offers

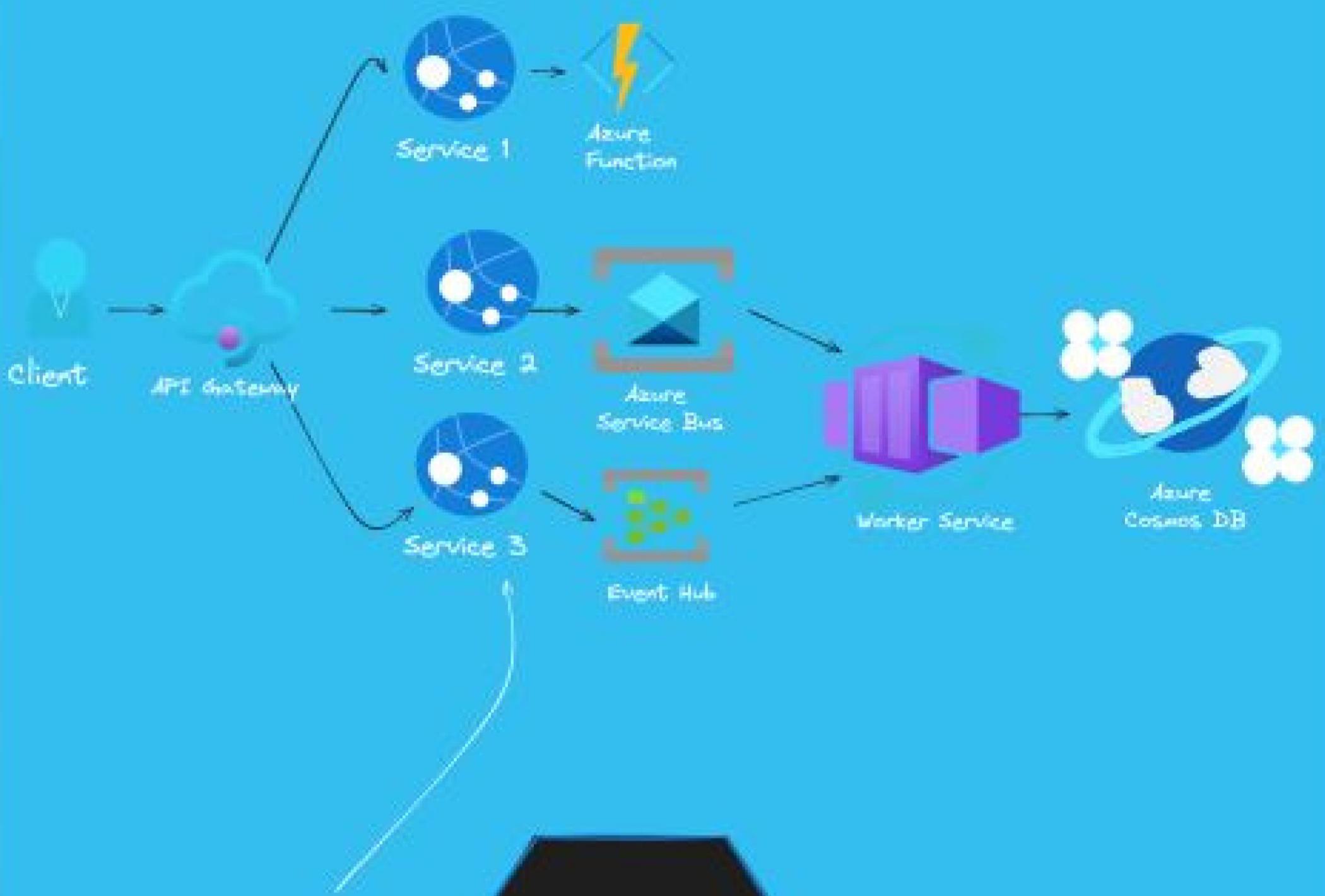
Documentation



IGOR IRIC



Ambassador Design Pattern



**I hope you find these tips
helpful!**

**Let me know if you have
any questions.**

Follow



Like



Comment



Igor Iric