FSI reference architecture based on Microsoft Azure

The content

- The problem statement
- Requirement analysis
- Solution design

The problem statement

“"Design a scalable and secure online banking platform that supports account management, payments, user authentication, and audit logging. Ensure high availability, modular design, and extensibility for future services."

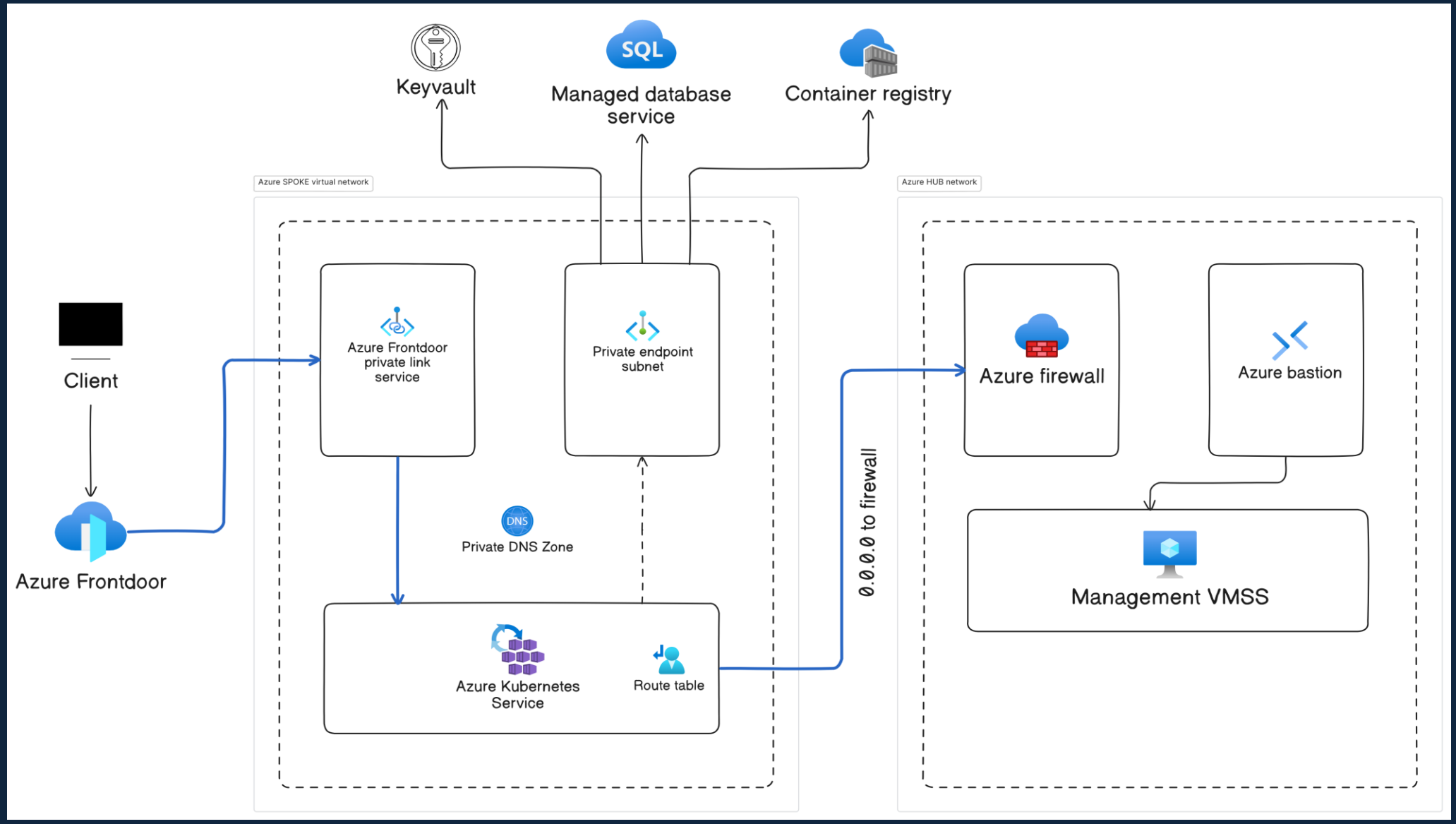
The identified functional requirements

- User account management (crud operations for accounts, view balance, etc.)
- Payments and transaction management (initiate payments, see transaction history)
- Authentication/Authorization (login, registration, password reset, MFA etc.)
- Audit logging to log critical events (append only log)

The identified non-functional requirements

- Scalability
  - Vertical and horizontal scaling
- High availability
  - Define target uptime (like 99.99 % SLA)
  - Disaster recovery plan (RPO, RTO minimums)
- Security and compliance
  - PCI-DSS, GDPR, PSD2 etc.
  - Regular audits
- Maintainability and modularity
  - Microservice or modular monolith architecture
  - Full blown CI/CD implementation
- Observability
  - Centralized logging and monitoring
  - Dashboard to monitor the health of the overall system
- Data management
  - Strong consistency model, backup/restore policies, data retention policies

The solution design (Azure single region)



The solution design

- Cloud-native solution based on Microsoft Azure (HUB and Spoke model)
- The application hosting platform is Azure Kubernetes Service
- Active/Active or Active/Passive multi region deployment
  - AKS multi-region-deployment
- DNS level load balancing with Azure Frontdoor, including WAF (Web application firewall)
- Azure Monitor + Application Insights for centralized logging and handling metrics
- Azure Confidential Ledger for append only audit logging
- The GitHub repository containing the documentation
  - https://github.com/attila-balogh-biro86/fsi-reference-architecture-azure

The solution design

- For authentication and authorization, the solution can leverage EntraID
  - https://learn.microsoft.com/en-us/azure/aks/azure-ad-rbac?tabs=porta
- Services like Azure Key vault provides a safe storage for secrets and certificates
- Microservice based approach let us scale the development teams/services according to the exact requirements
- Kubernetes HPA and Pod disruption budgets guarantee the high availability of the running service instances

Conclusion

- Azure Kubernetes based solution provides a scalable platform for multi-region highly available deployment
- Azure as a cloud vendor provides us the tools and compliance readiness which necessary to host a business-critical system like this
  - https://learn.microsoft.com/en-us/azure/compliance/
- As a starting point we can use modular monolith approach if the organization is not ready to a full blown microservice architecture
- With Express Route or S2S VPN we can easily integrate our on-premise infrastructure if there are any dependency on it.