**Context and Problem Statement**

The frontend application needs to communicate with each microservice in order to send and receive data to APIs.

**Decision Drivers**

* Method of access must be compatible with authentication and authorisation methods
* Must be scalable to accommodate increased traffic over time
* Services shouldn’t need to rely on each other

**Considered Options**

* API Gateway
* Backend For Frontend

**Pros and Cons of the Options**

**API Gateway**

* Good: Acts as a mask layer, hiding each microservice’s address from the internet
* Good: Using Yet Another Reverse Proxy (YARP), we can provide scalability by using it as a load balancer (*Load Balancing*, 2025)
* Good: Authorisation can be handle Upstream before passing request onto a service (*Authentication and Authorization*, 2025)
* Bad: Single point of entry for all devices. Data provided to the client can’t be catered to the client type
* Bad: Single point of failure. If the gateway was to go down, all access to the system will be lost

**Backend For Frontend**

* Good: Each Backend For Frontend is responsible for a specific client (Alireza Farokhi, 2022)
* Good: Able to communicate with different protocols such as GraphSQL, JSON
* Good: If one BFF were to go down, it wouldn’t prevent other devices from accessing the application
* Bad: Would take longer to implement
* Bad: Requires more maintenance due to multiple client requirements, increasing workload (Hislop, 2024)
* Bad: Not needed as the scope of the project is web only

**Decision Outcome**

As our application will be web only, we decided to go with a single API gateway. This would significantly reduce the overhead required to implement a gateway solution, and also require less maintenance on the frontend, due to only having one address to communicate with.

Authentication is easily implementable via YARP, allowing us to authorise requests before sending them to the targeted endpoint.

Further from this, having a gateway provides a layer of security by shielding each microservice from the internet as service IP addresses don’t need to provide their IP addresses to the clients (*What Is a Reverse Proxy? | Proxy Servers Explained*, 2024)

References

*Authentication and Authorization*. (2025). Github.io. <https://microsoft.github.io/reverse-proxy/articles/authn-authz.html>

Alireza Farokhi. (2022, November 23). *BFF Pattern vs Gateway Pattern - Alireza Farokhi - Medium*. Medium. <https://alirezafarokhi.medium.com/bff-pattern-vs-gateway-pattern-45706ffb9978>

*Load Balancing*. (2025). Github.io. <https://microsoft.github.io/reverse-proxy/articles/load-balancing.html>

*What is a reverse proxy? | Proxy servers explained*. (2024). Cloudflare.com. <https://www.cloudflare.com/en-gb/learning/cdn/glossary/reverse-proxy/>

Hislop, G. (2024, May 6). *The Pros and Cons of Using a Backend-for-Frontend (BFF)*. Medium. https://medium.com/@g.m.hislop93/the-pros-and-cons-of-using-a-backend-for-frontend-bff-a67e2edaefab