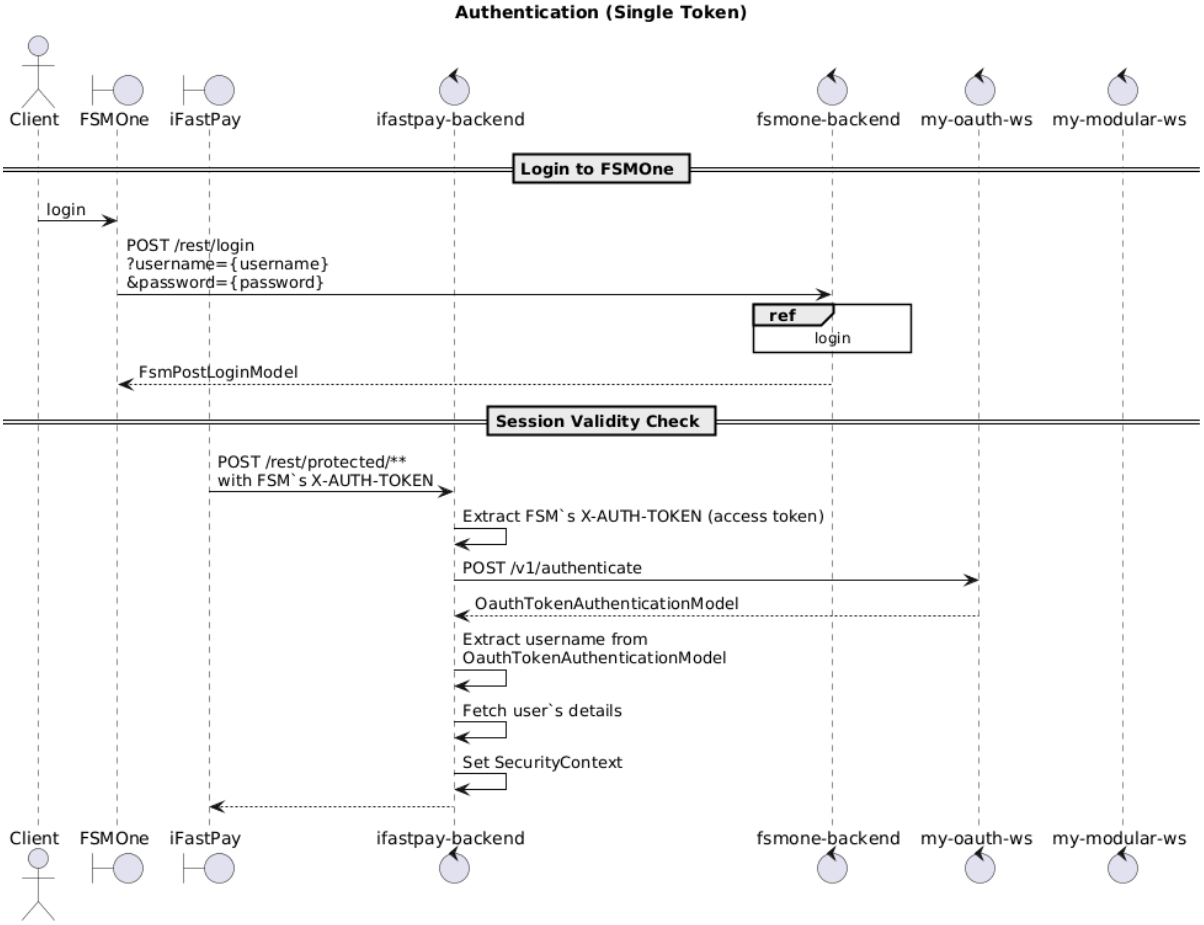
Approach 1 (Single Token)

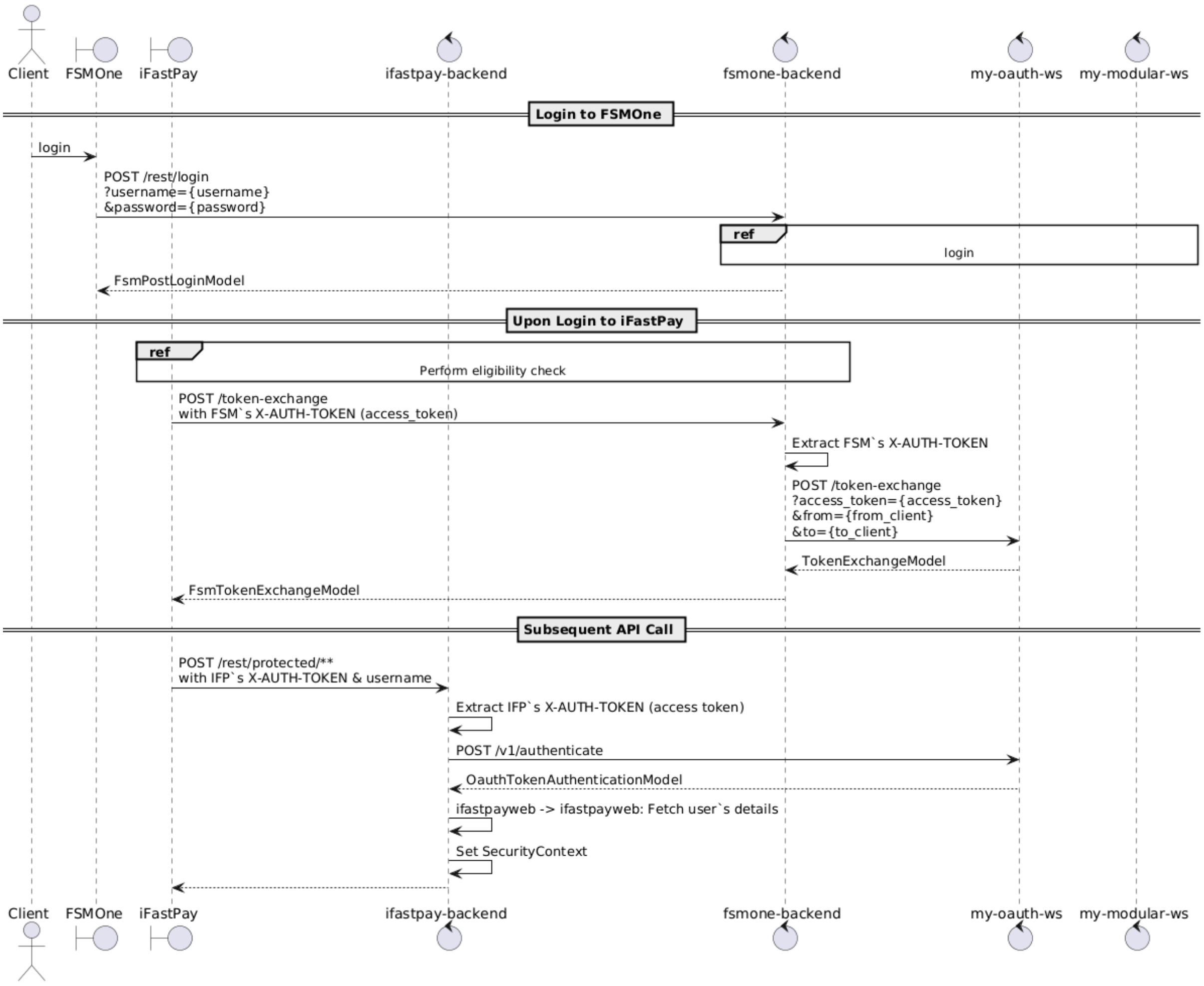


Notes (Single Token):

1. Both **FSMOne** and **iFastPay** share the same access token via the X-AUTH-TOKEN header when calling **my-oauth-ws** to validate the session.
2. The same token is valid for both FSM and iFastPay endpoints.

Approach 2 (Two tokens)

Authentication (Two Token)



Notes (Double Token):

1. From FSM, the **FSM backend** requests **my-oauth-ws** to perform a **token exchange**, obtaining a separate iFastPay access token.
2. At the end:
   a. **FSM token** can only be used for FSM endpoints.
   b. **iFastPay token** can only be used for iFastPay endpoints.
3. **Revocation**: Session revocation must apply to **all related tokens** (FSM and iFastPay).

Comparison

Aspect	Approach 1: Single Token (FSM)	Approach 2: Two Tokens (FSM & iFastPay)
Development Effort	Simple to develop — no major changes needed, minimal impact or the existing structure.	Harder to develop — requires changes in **my-oauth-ws** to support token exchange.
Complexity / Errors	Less error-prone, as fewer changes are involved.	More error-prone, since more changes are involved.
Token Storage	No change needed.	May require adjustments to token storage to support consistent revocatio
Future Flexibility	Future separation is more difficult.	Future decoupling is easier if iFastPay evolves into an independent system.
Audit Trail	Unclear — cannot determine which system is currently using the token.	Clear — audience is explicitly defined.
Least Privilege	Weaker — iFastPay inherits all access to FSMOne endpoints (and vice versa).	Stronger — least privilege by design, since each system uses a distinct toke with its own scope.