1. Authentication & Authorization

LX Stats:

Distinct members that logged into the system between 2025-01-01 and 2025-06-01: **456.819**

Distinct backend users that logged into the system between 2025-01-01 and 2025-06-01: **15.613**

Total: **472.432**

Comparison table of SAAS OAuth solutions:

Feature	Clerk	Azure AD B2C	Amazon Cognito	Firebase
Estimated Cost (472,432 users)	**$9.248,64/month**	**$1.372,91/month**	**$6.936,48/month**	**$2.112,16/month**
Free Users	**10,000 MAUs**	**50,000 MAUs**	**10,000 MAUs**	**50,000 MAUs**
MFA Pricing	**$100/month** for SMS, TOTP, backup codes	Charged per MFA attempt (SMS/Voice), pricing varies	**$0.01 per SMS message (inbound & outbound)**	**$0.005 per SMS message**
Ease of Use	Developer-friendly, simple setup	Requires Azure integration	Good for AWS users	Simple setup, great for mobile apps
Customization	UI components, flexible auth flows	Limited UI customization	Basic customization	Limited customization, but easy integration
Hosting	Fully cloud-hosted	Azure-only	AWS-only	Google Cloud-hosted
Security	MFA, SSO, role-based access	MFA, SSO, Azure security features	MFA, SSO, AWS security features	MFA, SSO, Google security features
Best For	Startups, modern apps	Microsoft ecosystem users	AWS ecosystem users	Mobile and web apps, startups

Required features:

Our authentication and authorization should follow the industry standards. There is no need to reinvent the wheel and create something non-standard that will be harder to secure, diagnose and expand.

OAuthServer Responsibilities

We'll keep using the OAuthServer we've already built and improved, since it follows OAuth2 standards and meets our needs without extra costs.

What it should do:

- **Handle logins and tokens** using OAuth2 and JWT.
- **Store user info**, like email, account name, and club-specific IDs.
- **Support multiple clubs** with clear data separation.
- **Control access** using roles per club and per service.
- **Let users log in** with a username and password. The username can be their email or something else they choose.
- **Offer 2FA** using apps (TOTP) or backup codes.
- **Share only the needed data**:
  - The central server manages shared login and role data.
  - Each service manages its own extra user data (like user settings or history).

Why we're doing this:

All the major OAuth providers work in similar ways and follow the same standards. Setting up one of them would take about the same effort as improving what we already have—but they'd cost us a lot more every month.

Since our own solution already works, sticking with it gives us more control and saves money. It's also built to work with how we handle multiple clubs and services.