

# Mastering Seamless Single Sign-On: Design, Challenges, and Implementation

RINKU MOHAN

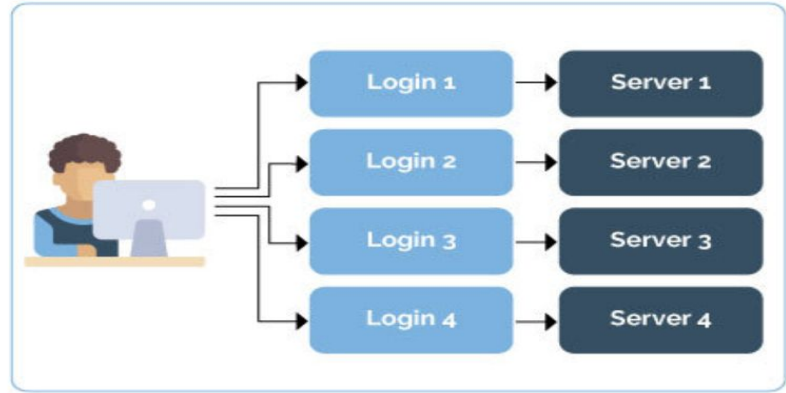




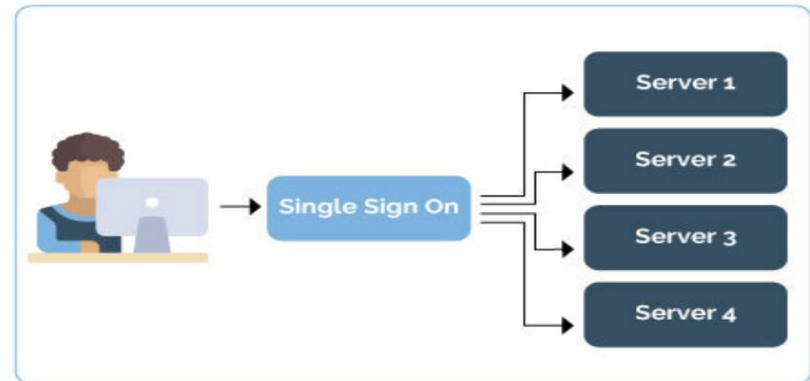
# Why Do We Need SSO?

- **Before SSO:** Users had to log in separately to each application, leading to **password fatigue**, **security risks**, and **inefficiencies**.
- **With SSO:** Users authenticate once and gain access to multiple services seamlessly.

## Without Single Sign On (SSO)

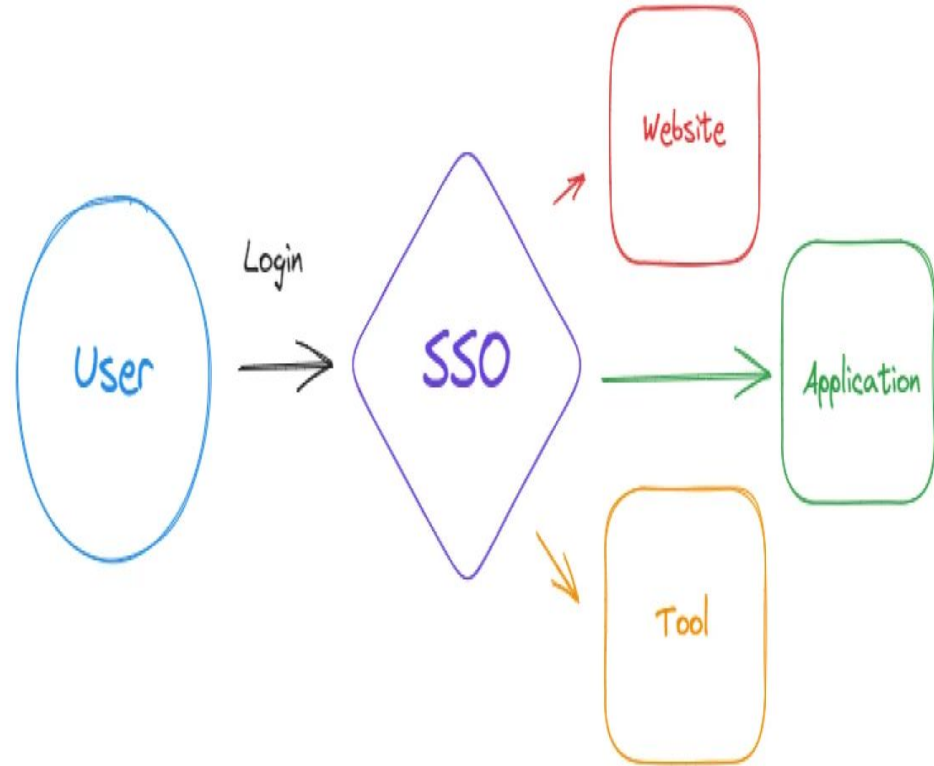


## With SSO



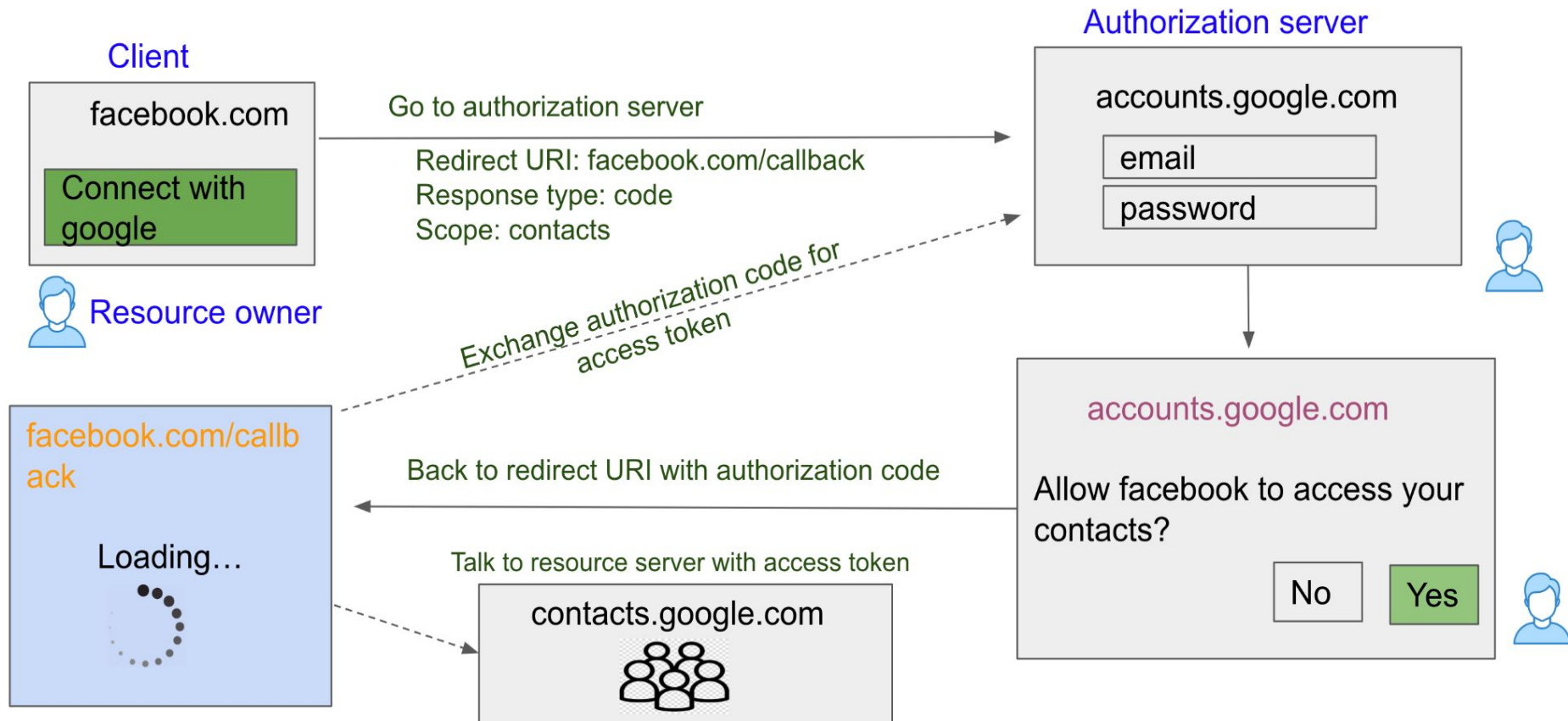
# Core SSO Protocols

- **OAuth2** (Token-based, modern web & mobile apps)
- **SAML** (XML-based, legacy enterprise systems)

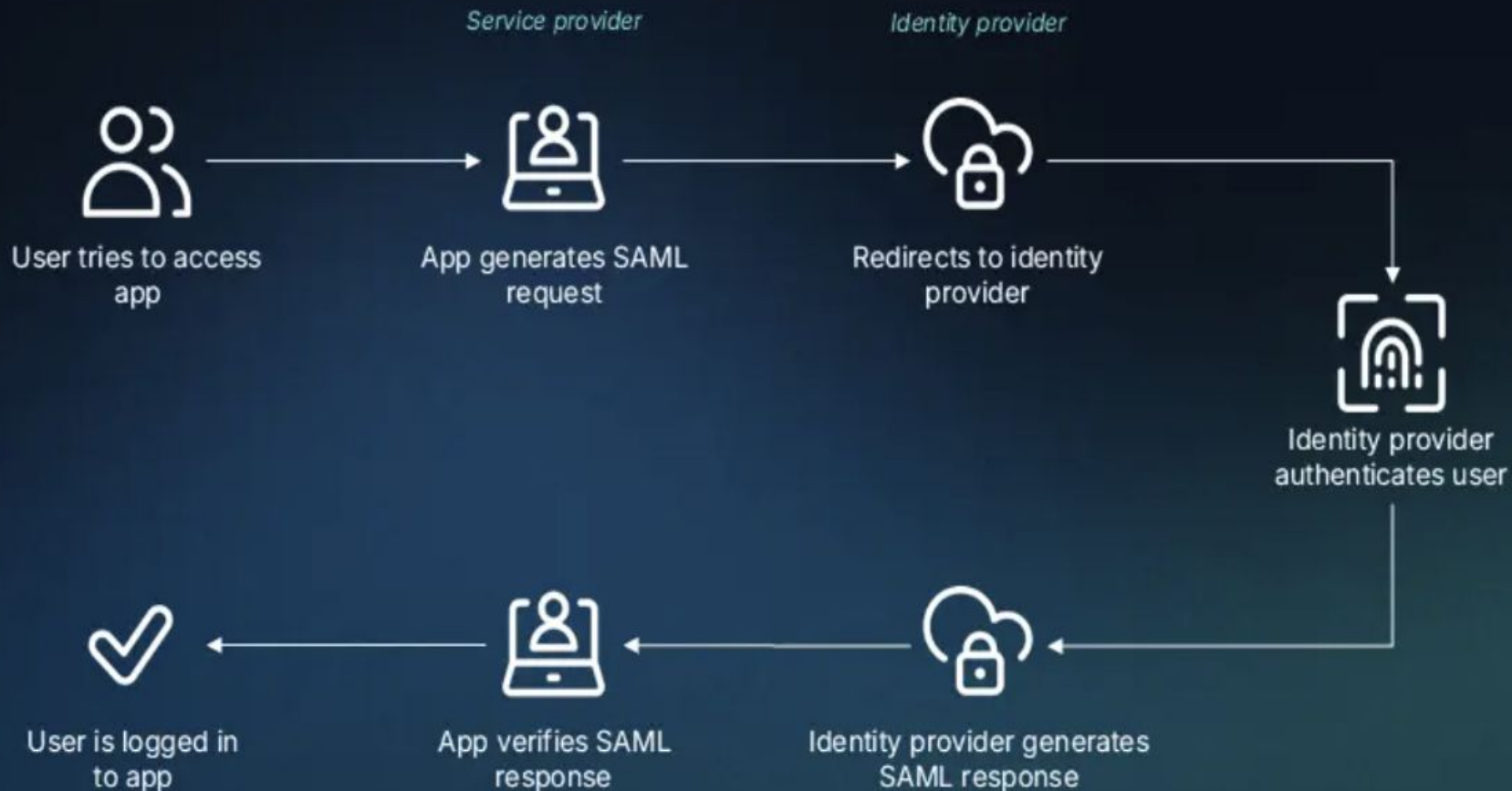




# OAuth 2.0 authorization code flow



# SAML SSO Authentication



# Designing a Seamless SSO System

Balancing **Security**, **Performance**, and **User Experience** is key.

1. **Security:** Protect tokens, enable MFA.
2. **Performance:** Optimize token validation, cache responses.
3. **User Experience:** Keep login flows intuitive.





# Security Challenges in SSO

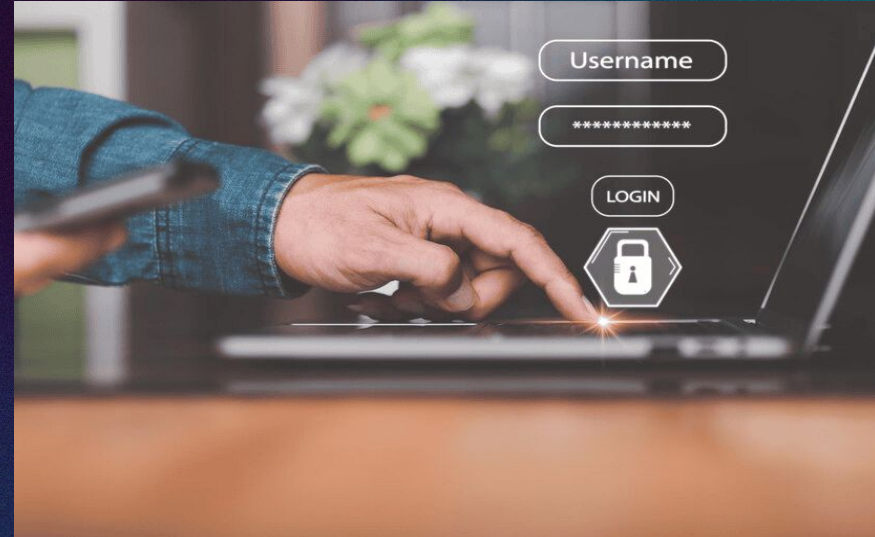
- **Single Point of Failure:** If your Identity Provider (IdP) is compromised, attackers get broad access.
- **Token Theft & Replay Attacks:** Attackers can steal and reuse tokens.
- **Man-in-the-Middle (MITM) Attacks:** Intercepting token exchanges.





# Best Practices for Secure SSO

- Use **short-lived tokens** & refresh tokens.
- Enforce **Multi-Factor Authentication (MFA)**.
- Implement **token scoping** (limit permissions per token).
- Enable **logging & monitoring** for anomaly detection.





# Performance Optimization

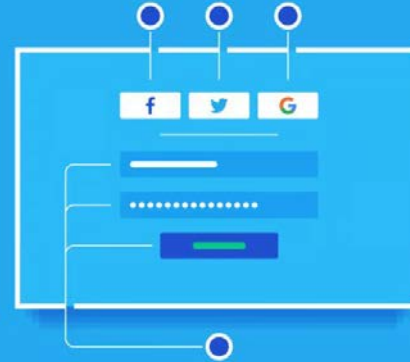
- Cache authentication tokens to **reduce backend load**.
- Load balance **identity providers** to handle high traffic.
- Use **async token validation** to speed up processing.





# UX Considerations for SSO

- Minimize login prompts.
- Provide **branded login pages**.
- Ensure **session persistence** for better experience.





# Integrating with Legacy Systems

- Use **SSO gateways** to wrap legacy apps.
- Gradually replace outdated authentication mechanisms.
- Test compatibility before full rollout.





# Future of SSO & Authentication

- **Zero Trust** authentication models.
- **AI-powered anomaly detection.**
- **Decentralized identity**  
(Blockchain-based authentication).





# Key Takeaways

- SSO improves security & UX but must be carefully designed.
- Choose the right protocol (OAuth2 vs. SAML).
- Balance security, performance & usability.
- Optimize token handling & authentication workflows.





# THANK YOU

RINKU MOHAN

<https://www.linkedin.com/in/rinku-mohan-06419310a/>