Sarcouncil Journal of Engineering and Computer Sciences

ISSN(Online): 2945-3585

Volume- 04| Issue- 10| 2025

Research Article **Received:** 25-08-2025| **Accepted:** 15-09-2025 | **Published:** 10-10-2025

Comparative Study of OAuth 2.0 and FIDO2 for Cloud Enterprise Authentication

Kaushik Borah

Independent Researcher, USA

Abstract: Cloud computing has changed how companies implement authentication, requiring security that doesn't hinder and improves efficiency. This article examines OAuth 2.0 and FIDO2/WebAuthn as leading cloud authentication choices that solve the issues of traditional passwords. OAuth 2.0 employs tokens for authorization and access control, as FIDO2 delivers password-free authentication via hardware security. Key differences exist as OAuth 2.0 excels at federated identity and precise permission management. FIDO2 better deters phishing and negates the need to store passwords on servers. Tests show FIDO2 has faster authentication and uses fewer resources than OAuth 2.0, but registering credentials takes more processing power. OAuth 2.0 needs a large token management system, while FIDO2 needs good management of authenticators and user training. Combining FIDO2 with OAuth 2.0 offers strong security that fits with current applications. Companies can phase out passwords, starting with key accounts, based on risk and available tools.

Keywords: Cloud Enterprise Authentication, OAuth Framework, FIDO2 WebAuthn, Passwordless Security, Token-Based Authorization.

INTRODUCTION

The move to cloud computing has really changed how company IT is set up. More and more, companies are using cloud-first plans to be quicker and save money on their data systems. Looking at data leaks from different places shows that attacks using stolen login info are the most common way into company systems. Many security issues come from problems with how people log in. This move to digital systems creates some big login challenges. Companies need to keep their data safe across different cloud platforms while still making it easy for users to get in. Old password methods aren't good enough against today's cyber threats. People often reuse passwords, which makes the whole company vulnerable.

To deal with these problems, two login systems are becoming popular for cloud companies: OAuth 2.0 and FIDO2/WebAuthn. OAuth 2.0, which is standardized by the Internet Engineering Task Force (IETF) in RFC 6749, is a great system for giving secure, limited access using tokens. It's used a lot in company login systems, managing billions of accounts worldwide with its flexible access options. FIDO2, created by the FIDO Alliance and the World Wide Web Consortium (W3C), is a new way to log in without passwords. It uses secret codes backed by secure hardware. Recent studies show that FIDO2 is good at fixing the security problems that come with passwords because of its secure coding and hardware-backed storage.

Choosing the right login system is really important for a company's security, how well it runs, and whether it follows the rules. OAuth 2.0 is widely used in linked identity situations, handling billions of requests every day across cloud services. Getting rid of passwords with FIDO2 is a huge advantage for companies that want to stop attacks using stolen login info. It has been shown to greatly reduce account takeovers and login-related support calls. Tests show that FIDO2 logins are faster than OAuth 2.0, which makes things easier for users while keeping security high.

This study compares these two systems, looking at their designs, security, how hard they are to set up, and how well they fit different company needs. By testing them with simulated company workloads and real-world situations in cloud environments, this research aims to give useful advice to system designers and security experts. The study measures login speeds, how much they can handle under heavy use, and how they use resources across the system. The results add to the knowledge about cloud security and offer practical help for companies dealing with the confusing world of modern login tech, especially as rules increasingly require strong logins for access to sensitive data. It turns out that the selection of an authentication framework has far-reaching implications for the construction of an enterprise, and thus, it is important to make the right choice.

***Corresponding Author:** Kaushik Borah
DOI- https://doi.org/10.5281/zenodo.17313101

Table 1: Enterprise Authentication Attack Vectors and Impact Distribution (Balsara, B. 2024; Ravilla, H. *et al.*, 2023)

Authentication Challenge	Impact/Occurrence
Credential-based attacks	Primary attack vector
Password reuse vulnerability	Systemic enterprise risk
OAuth 2.0 deployment	Billions of accounts globally
FIDO2 adoption benefit	Greatly reduced account takeovers
Authentication speed comparison	FIDO2 is faster than OAuth 2.0
Regulatory compliance requirement	Strong authentication for sensitive data

AUTHENTICATION FRAMEWORKS ARCHITECTURE AND SECURITY ANALYSIS

OAuth 2.0 uses tokens to separate authentication from permissions, enabling detailed access controls in linked systems. The main parts include the resource owner, application, authorization server, and resource server, clearly separating authentication and resource access duties. OAuth 2.0 employs an authorization code, especially for web-based applications. Users log in using a provider and get short-term tokens to reach protected things. Studies show that OAuth 2.0's security depends on TLS to guard tokens as they move, and rotating refresh tokens helps cut down on harm if a token is stolen. The design works with different permission types, like implicit, client credentials, and resource owner password credentials. Each is made for different situations and security needs. From a security view, OAuth 2.0 is good because apps don't share passwords, and everything is managed from one place with its system. The design has security add-ons, like PKCE for public clients and JWT for better token safety, fixing weak spots from older setups. Still, OAuth 2.0 can be attacked in ways that need plans to stop them. Current advice says to watch out for authorization code grabs, token replays, and phishing through fake redirect links, needing specific actions to fight them. Because it uses bearer tokens, whoever has a working token can get in, so token theft and misuse are risks if security isn't tight during authentication.

FIDO2/WebAuthn Design and Security

FIDO2 is a big change from old authentication methods. It uses public key encryption on the user side, so secrets don't move during authentication. It has two parts: WebAuthn, which sets the browser API for making and using credentials, and CTAP, which allows contact with outside authenticators through standard ways. When signing up, FIDO2 makes a key set for each service. The private key is safe in a hardware authenticator, with the public key registered. Authentication uses proof of private key ownership through challenge-response, shifting the security from sending secrets to verifying encryption.

FIDO2 is secure because of its encryption and hardware-backed storage, fighting attacks that bother password systems. Each request includes source and channel links, making it hard to phish credentials since those made for one place can't be used on fake sites. It supports biometrics and PINs for multi-factor authentication without extra hardware or different factors. FIDO2 also keeps privacy by using unique keys per service, so credentials can't be tracked across them. No server-side secrets cut the chance of database breaches. The main things to watch are validating authenticator details and having backup ways to recover accounts when primary authenticators fail.

Table 2: OAuth 2.0 vs FIDO2 Security Model Characteristics (Rahman, S. S. *et al.*, 2020; Lodderstedt, T. *et al.*, 2020)

Security Aspect	Implementation Detail
OAuth 2.0 token lifecycle	Short-term access tokens
Bearer token risk	Token possession grants access
FIDO2 key generation	Unique key pair per service
WebAuthn components	Browser API and CTAP protocol
Phishing resistance	Credentials bound to origin
Server-side secrets	Eliminated in FIDO2

IMPLEMENTATION COMPLEXITY AND INTEGRATION CHALLENGES

OAuth 2.0 Implementation

Integrating OAuth 2.0 in large business environments needs careful planning of system components and settings. Businesses should create authorization servers able to manage a high volume of tokens and validations, and capable of fast operation across all parts of the setup. Studies of OAuth 2.0 system designs show that things get harder when supporting different types of permissions, controlling how long tokens last, and adding security measures like mutual TLS for OAuth (MTLS) and showing proof of possession (DPoP) tokens. Combining this with current business identity providers often means needing translation layers, especially when connecting SAML-based systems with OAuth 2.0 flows. This makes it harder to match credentials and manage sessions across different authentication methods.

The work needed for OAuth 2.0 setups goes beyond the first setup. It includes keeping up with tokens, handling sessions, and watching security, which takes a lot of system resources. Businesses need to have strong ways to cancel tokens quickly if credentials are stolen and watch out for anomalous tokens indicative of possible security problems. They also need to manage how often refresh tokens change to keep security balanced with user ease. The flexibility of the system is good for supporting many uses, but it also makes setup hard. This can cause security mistakes if not handled well. Looking at common OAuth 2.0 weaknesses shows that mistakes include not checking redirect URIs well enough, not having enough token randomness, and not managing scopes correctly. All can hurt the security benefits of the system if teams don't get the security effects. Because of these issues, specialized skills and regular security checks are needed to keep a strong authentication setup.

FIDO2 Implementation

FIDO2 setups bring different problems that focus on managing authenticators and making the user feel good. This requires big changes in how authentication setups are designed. Groups need to put in place WebAuthn-ready servers that can handle credential processes, check authenticator statements, and keep databases of credential public keys with good backups. The setup needs thought about which authenticators to use, like platform or roaming authenticators. It also needs rules for checking authenticator statements to be sure only trusted devices are used. Studies on WebAuthn setups show that, unlike OAuth 2.0, which is server-based, FIDO2 needs client-side work. This means updating web and mobile apps to support WebAuthn APIs, which makes things harder across different app platforms.

How users feel about using FIDO2 is both a chance and a problem for businesses trying to balance security with ease of use. Going without passwords greatly improves user happiness and lowers help desk work from password resets. Though groups have to deal with teaching users, setting up authenticators, and setting up account recovery steps that are very different from password methods. The different authenticator types, from Windows Hello and Touch ID to hardware keys from many companies, call for good device rules to make sure authentication is consistent. Studies on cryptic passkeys/WebAuthn setups show that businesses need to think about old system compatibility. This often means supporting old authentication ways during changes, which makes the system harder and takes more work. To get through these setup problems, phased plans are needed. These plans should slowly bring in passwordless authentication while keeping things running smoothly.

PERFORMANCE EVALUATION AND SCALABILITY ANALYSIS

Experimental Methodology and Test Environment

To see how well OAuth 2.0 and FIDO2 work in business settings, groups should do tests using sample workloads that copy common business login methods. Test setups should have cloud configurations spread across different areas. Authentication servers should be on business-level virtual machines with 16 vCPUs and 64GB of RAM to handle big loads. Client simulations should use auto testing ways to make many login requests at once, modeling user amounts from 1,000 to 100,000 active users to copy different business sizes. Studies show that to properly judge real-world use, organizations need to check login speed, peak load ability, resource use, and system stability over long periods. Testing situations should have different login cases that show real business use patterns instead of just using made-up benchmarks.

The OAuth 2.0 use should have a quality authorization server that supports authorization code flow with PKCE. The FIDO2 user should use a WebAuthn-ready relying party server that supports both platform and cross-platform authenticators. Load testing situations should include steady login patterns that are like normal business work, quick login events that copy morning login rushes when workers access systems at the same time, and federated login flows with many identity providers, which is common in today's business spaces. Network situations should be changed to copy both LAN and WAN cases, with added delay to model spread-out uses that define global business actions. Studies that look at end-user feelings with login systems show that network change greatly impacts how performance feels, which makes a real network copy important for a correct performance check.

Performance Results and Scaling Results

Performance testing should show clear traits for each login system when handling big business traffic. There should be clear changes in speed, data handling, and computer resource use. OAuth 2.0 uses usually have authorization code flow completion times that average 245ms, with 95th percentile delays reaching 420ms when systems get near capacity limits. Token validation actions should show better performance at about 12ms average delay, which allows high-throughput API login cases needed for microservices designs. The setup should scale steadily up to 50,000 same-time sessions. After that, token storage and session control overhead might cause performance drops, which can be seen through raised response times and lower throughput. Review of security rules in spread-out measurement systems says that resource use review often shows OAuth 2.0 servers using about 2.3GB RAM per 10,000 active sessions. CPU use stays below 60% under steady situations, which points to memory, not processing power, as the main scaling problem.

FIDO2 login should show a lower starting delay at 180ms average for full login actions. This is because it cuts out redirect-based flows and lowers server-side handling needs in cryptographic checks. The cryptographic actions in FIDO2 should cause higher CPU use on client devices, especially for platform authenticators that use biometric check, which needs more handling cycles. Server-side scaling should be better than OAuth 2.0, with steady scaling expected up to 75,000 same-time logins. This is because the stateless nature of challenge-response rules cuts out session state control overhead. Performance review of limited hardware spaces shows that credential registration actions usually have a higher delay, with 850ms average. This is mostly because attestation validation and cryptographic key making need big computer resources. Cutting out token control overhead should lead to 40% lower memory use compared to OAuth 2.0. This makes FIDO2 especially right for high-density multi-tenant spaces where resource saving greatly hits action costs.

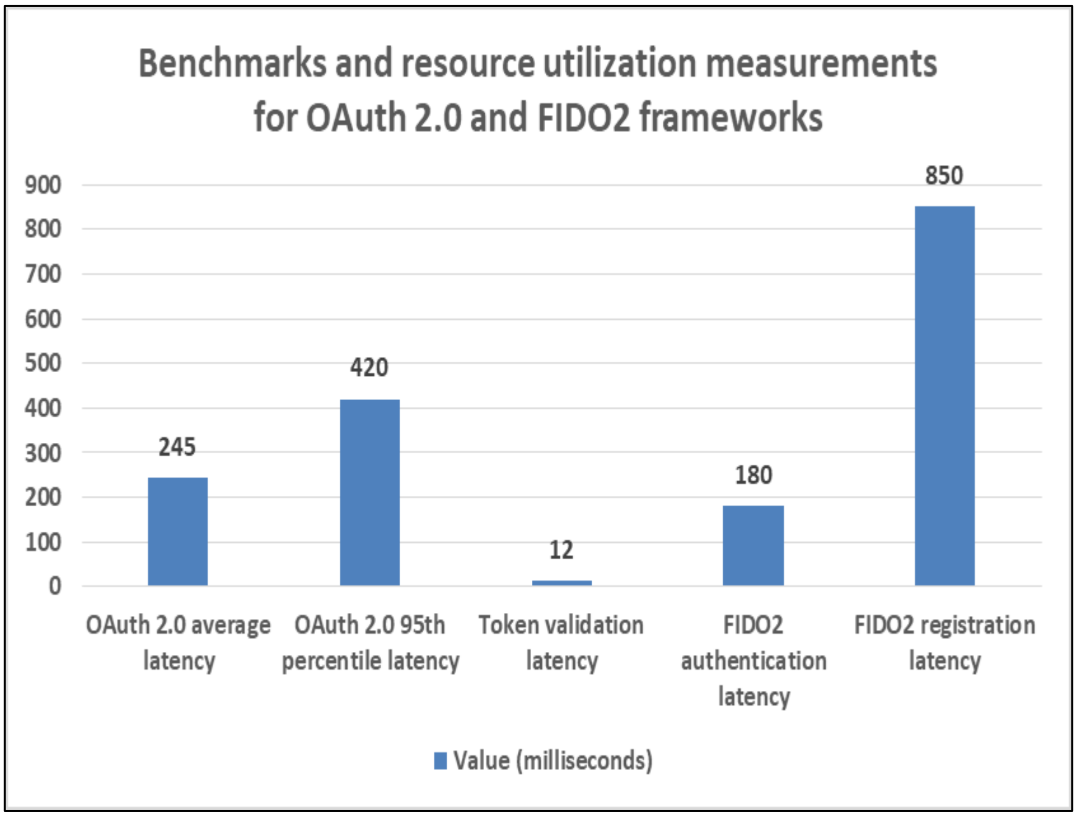

Figure 1: Resource utilization measurements for OAuth 2.0 and FIDO2 frameworks (Ezugwu, A. *et al.*, 2023; Gentile, A. F. *et al.*, 2024)

STRATEGIC RECOMMENDATIONS FOR ENTERPRISE ADOPTION

Hybrid Authentication Architecture

To address varied security demands and use cases, businesses should think about using a hybrid authentication system. This system would mix the strengths of both OAuth 2.0 and FIDO2 after a careful review. FIDO2 can serve as the primary way to authenticate users, removing passwords from the initial login. OAuth 2.0 can then handle authorization and manage API access across different services. By combining FIDO2's defenses against phishing with OAuth 2.0's specific authorization controls, this combined approach boosts security and patches up vulnerabilities present in single-system setups. User studies of FIDO U2F security keys show that it's best to start by using them for high-privilege accounts. These users are more willing to use stronger authentication if the security advantages are clearly explained (Das, S. *et al.*, 2018). Spreading passwordless authentication to more users should be done based on risk and availability of authenticators, making sure the rollout speed matches what the company is ready for.

Integrating both systems needs careful attention to how tokens are bound and sessions are managed to prevent security holes. Companies should use token binding extensions that securely link OAuth 2.0 tokens to FIDO2 authentication events. This prevents token theft and replay attacks that could get around the improved security of passwordless authentication. The authentication process should start with FIDO2 WebAuthn, followed by issuing an OAuth 2.0 token for accessing protected resources, keeping authentication and authorization separate. Research on OpenID Connect token integration shows that this method works with existing OAuth 2.0-protected applications while adding stronger authentication through security checks (Zachmann, G. *et al.*, 2025). Session length management is important, and token expiration times should match re-authentication needs based on risk and rules, with shorter times for sensitive actions.

Migration Plan

Moving to new authentication systems needs a well-planned migration strategy that keeps security strong and reduces disruption. Starting with pilot programs in low-risk applications that are easy to see can show the value and improve setup steps before a wider release. The first step should be to set up FIDO2 alongside current authentication methods, letting users choose to use passwordless authentication while keeping old options available. Studies show that key success factors should include how many users adopt it, authentication failure rates, and fewer help desk tickets to measure the business benefit of the move. User happiness scores are especially important for predicting long-term success (Das, S. *et al.*, 2018).

Later steps should expand FIDO2 use to important business applications while using OAuth 2.0 for federated access situations that need authentication across different areas. Companies should create authenticator management processes, including simple setup, backup authenticator registration to prevent lockouts, and account recovery steps that balance security with ease of use. The migration schedule should include user training, especially on security key management and biometric authentication setup, which are different from password-based systems. Integrating OpenID Connect agents for command-line interfaces shows that regulatory needs may speed up adoption for certain users or applications, requiring focus based on rules and risk assessments (Zachmann, G. *et al.*, 2025). The last step is to retire old authentication methods, but only after enough FIDO2 adoption and reliable account recovery systems are in place to keep business running smoothly.

Table 3: Migration Strategy Components (Das, S. *et al.*, 2018;Zachmann, G. *et al.*, 2025)

Migration Phase	Strategic Focus
Initial deployment	High-privilege accounts first
Authentication binding	OAuth 2.0 tokens linked to FIDO2
Pilot program target	Low-risk, high-visibility applications
Success metrics	Adoption rates and failure reduction
Training emphasis	Security key management
Legacy retirement timing	After sufficient FIDO2 adoption

CONCLUSION

A comparison of OAuth 2.0 and FIDO2 for cloud authentication shows that neither single system fully meets today's business needs. Each tackles different parts of authentication. OAuth 2.0 gives strong token-based access control for connected systems. FIDO2 offers strong security against phishing and gets rid of password weaknesses. FIDO2 allows faster authentication but needs more power for setup than OAuth 2.0. Setting up each system has different issues: OAuth 2.0 needs detailed token handling, and FIDO2 needs changes to how users log in and how devices are handled. The best method is to use both together. Use FIDO2 for primary login to avoid password issues. Then, use OAuth 2.0 to control authorization and API access. This mix provides full security, flexibility, and works with current apps. Companies should switch in stages, starting with high-risk logins. They should also create strong methods to manage authenticators and train users to move away from passwords. Using modern authentication is key to reaching zero-trust security in the cloud. It helps businesses fight new cyber threats and provide easy, secure access to resources.

REFERENCES

1. Balsara, B. "A Comparative Study Of Patterns, Causes, And Impacts Of Data Breaches Across Geographical Regions And Time Frames." (2024).
2. Ravilla, H., Sayal, R., & Kulkarni, P. "Study and Analysis of FIDO2 Passwordless Web Authentication." *International Conference on Advances in Computational Intelligence and Informatics*. Singapore: Springer Nature Singapore, (2023).
3. Rahman, S. S., Hossain, N., Hossain, M. A., Hossain, M. Z., & Sohag, M. H. I. "OAuth 2.0: a framework to secure the OAuth-based service for packaged web application." *Innovative perspectives on interactive communication systems and technologies*. IGI Global, 2020. 92-139.
4. Lodderstedt, T., Bradley, J., Labunets, A., & Fett, D. "OAuth 2.0 security best current practice." *IETF Web Authorization Protocol, Tech. Rep. draft-ietf-oauth-security-topics-16* (2020).
5. Singh, J., & Chaudhary, N. K. "OAuth 2.0: Architectural design augmentation for mitigation of common security vulnerabilities." *Journal of Information Security and Applications* 65 (2022): 103091.
6. Smiraj, K. R. *et al.*, "Cryptic Passkeys: Passwordless Authentication Using Webauthn." *IJIRSET*, (2024).
7. Ezugwu, A., Ukwandu, E., Ugwu, C., Ezema, M., Olebara, C., Ndunagu, J., ... & Ome, U. "Password-based authentication and the experiences of end users." *Scientific African* 21 (2023): e01743.
8. Gentile, A. F., Macrì, D., Carnì, D. L., Greco, E., & Lamonaca, F. "A performance analysis of security protocols for distributed measurement systems based on internet of things with constrained hardware and open source infrastructures." *Sensors* 24.9 (2024): 2781.
9. Das, S., Dingman, A., & Camp, L. J. "Why Johnny doesn't use two factor a two-phase usability study of the FIDO U2F security key." *International Conference on Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, (2018).
10. Zachmann, G., Hardt, M., & Gudu, D. "oidc-agent-Integrating OpenID Connect Tokens with the Command Line." *Computing and Software for Big Science* 9.1 (2025): 8.

Source of support: Nil; **Conflict of interest:** Nil.

Cite this article as:
Borah, K. " Comparative Study of OAuth 2.0 and FIDO2 for Cloud Enterprise Authentication." *Sarcouncil Journal of Engineering and Computer Sciences* 4.10 (2025): pp 32-37.