Human Systems Engineering and Design (IHSED2025), Vol. 198, 2025, 103–112
https://doi.org/10.54941/ahfe1006766

Securing Interfaces of a Multinational Standard With Technical Specifications for Data Sharing: Challenges of Authentication and Authorization

Lorraine Hagemann, Simon Schwarz, and Barbara Essendorfer

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany

ABSTRACT

Standards are helpful to establish interoperability within multinational coalitions. In a military context the NATO standard STANAG 4559 outlines models and processes for the sharing of Intelligence, Surveillance and Reconnaissance (ISR) data. This paper explores the intricate challenges of securing such data dissemination processes, particularly focusing on authentication and authorization mechanisms. As security requirements evolve from a "System High" to a "Zero Trust" approach, the need for stringent identity verification and privilege management becomes paramount, especially in untrusted network environments. We analyze various authentication and authorization technologies, from Basic Auth to OpenID Connect (OIDC), to identify their applicability within the constraints of a multinational data sharing standard. We highlight key challenges, including compatibility with legacy systems, coordination for common (that is, compatible) configurations, and the implications of integration within a broader network context. Through empirical case studies and participation in exercises, we provide insights into effective strategies for overcoming these obstacles, thereby contributing to the development of robust security frameworks in coalition operations.

Keywords: STANAG , Coalition shared data, Security, Authentication, Authorization

INTRODUCTION

Multiple nations participate as coalition partners in a Joint ISR (Intelligence, Surveillance and Reconnaissance) enterprise according to the Coalition Shared Data (CSD) concept, that is specified in a NATO standard. To be able to collaborate in a multinational network and share data robust security measures are essential, and secure data exchange must be guaranteed.

The requirements for data sharing have evolved from a "System High" approach with implicit trust on the network to the design paradigm "Zero Trust".

Among other aspects enforcing authentication (asserting one's identity) and authorization (asserting one's privileges) are essential components of ensuring secure communication in a Zero Trust environment. Several widespread technologies that provide authentication and/or authorization

© 2025. Published by AHFE Open Access. All rights reserved.

for HTTP-based communication in a network exist. These range from simple solutions like Basic Auth to complex technologies like OIDC (OpenID Connect). There is a whole ecosystem of either free or paid service providers that offer to take some of the burden from the actual services by centralizing authentication and authorization in their own identity provider service. However, technical as well as organizational challenges arise when a network does not have a central controlling entity. This often is the case when implementing CSD systems and services in a coalition provided by different vendors and nations.

In this paper we examine the complexities of securing interfaces within a multinational standard for data sharing, with a particular focus on the challenges of authentication and authorization. We discuss the issues that arise in such use cases and how to adapt common technical solutions to them, the trade-offs involved and the applicability in a multinational standard based context. We also present some empirical experiences and insights gathered during exercises.

The paper is structured as follows: In the chapter Coalition Shared Data we briefly introduce the CSD context. In the chapter Challenges of Authentication and Authorization we explain the key challenges of securing interfaces by appropriate authentication and authorization mechanisms. The chapter Authentication and Authorization Mechanisms introduces several authentication and authorization mechanisms and explores their applicability for STANAG 4559. After the evaluation of the different variants, we introduce our choice in the chapter Implementation Experience. Additionally, we lay out our experiences testing our solution. Finally, the paper concludes with a summary of our results and experience and an outlook on future work.

COALITION SHARED DATA

The CSD concept specifies the interoperable data exchange in multinational Joint ISR operations. Here multiple heterogeneous services, servers and clients provided by different organizations from different countries (potentially) using different technologies and influenced by unique governance structures all need to communicate in an interoperable way. The technical specifications to support this concept are described in the NATO standard STANAG 4559 (NATO Standardization Office (NSO), 2018a). It defines a data model, services and interfaces to exchange (military) intelligence data to support the Joint ISR process. Such a standard enables interoperability between coalition partners and prevents technical incompatibilities. The standard is organized in three AEDPs (Allied Engineering Documentation Publications) that focus on different types of data and processes. In our paper we focus on CSD servers according to AEDP-17 (NATO Standardization Office (NSO), 2018b) that enable the sharing of Joint ISR products as exploited imagery, documents and reports. As depicted in Figure 1, client systems are connected to a CSD server to store or retrieve data. Multiple CSD servers are connected with each other to share that data. However, in operational use cases CSD servers are usually

integrated in a greater deployment within a network of networks approach. Besides the server-to-server interfaces to exchange data with other CSD servers, client/user-facing interfaces are also available to store and catalogue data and be able to query for data.

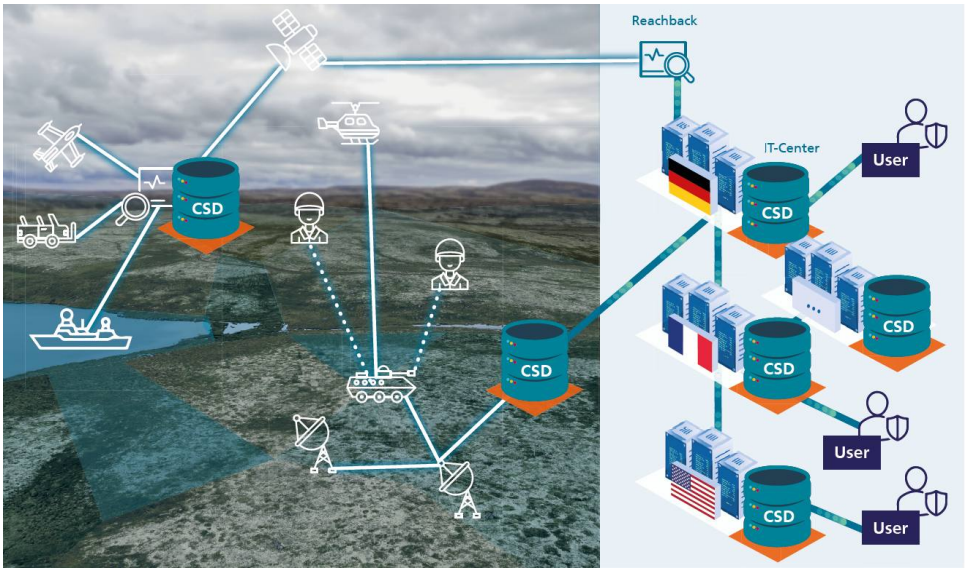

Figure 1: Exemplary CSD network with multiple CSD servers from different nations connected with different network types.

In summary, use cases where different systems and services are connected to each other and thus authentication and authorization is of relevance involve server-server interaction as well as client-server interaction.

The origin of STANAG 4559 lies in the early 2000s. Since then it has been updated and improved according to the latest developments of IT standards and especially IT security requirements, as well as functional needs. Since various international partners are involved and interoperability must be guaranteed, changes to the standard cannot be made without considering all stakeholders. Therefore, continuous maintenance and development of the standard is coordinated within a Custodian Support Team (CST) (Rodenbeck et al., 2024).

As we outlined in earlier work (Hagemann & Klotz, 2024) in the origins of STANAG 4559 it was assumed that systems would operate in protected networks and thus implicit trust between the systems and servers of the coalition partners offered sufficient protection (System High approach). Therefore, according to the latest version of AEDP-17 (Edition A Version 3) authentication and authorization is solely an optional vendor-dependent feature without precise guidelines. Considering state-of-the-art security requirements, the need for a Zero Trust architecture has emerged. This means, in contrast to an implicit trust approach, dedicated security measures are required for all levels. Therefore, all communication must be

secured with suitable mechanisms, including mechanisms for authentication and authorization.

CHALLENGES OF AUTHENTICATION AND AUTHORIZATION

When selecting appropriate authentication and authorization mechanisms, a range of requirements (legal, organizational and technical) must be considered. In addition to reaching an agreement within the CST, it is also necessary to consider the greater picture, namely overarching architectures, processes and networks. Below, we present the key challenges associated with the introduction of authentication and authorization within the CSD context.

a) Compatibility with legacy systems: Due to the long existence of STANAG 4559, it is plausible that certain servers, referred to as legacy systems, may not incorporate the latest modifications. Consequently, these legacy systems could be present, particularly in existing deployments. Moreover, even in new deployments, coalition partners may integrate a legacy system. Therefore, it is essential that changes to the specification are designed to be backward compatible whenever feasible.

b) Compatibility with specification in the standard: The current version of the standard defines interfaces and services implemented with specific technologies (e.g. SOAP). Consequently, by specifying the auth layer, it is essential to choose a mechanism that is either inherently compatible or can be modified to work with the particular technologies prescribed by the standard.

c) Coordination for compatible configurations: The coalition network comprises systems and servers from multiple coalition partners, specifically from various vendors. To ensure a smooth and interoperable process, it is essential that configurations, such as network settings and synchronization configuration, are harmonized. Ideally, a centralized entity tasked with coordinating these configurations would be advantageous. However, the diversity among coalition partners raises the question of who would be responsible for managing this central authority. Naturally, coalition partners favor maintaining their autonomy and thus a decentralized mechanism. The complexity of coordinating unified configurations can vary significantly, depending on the chosen authentication and authorization mechanisms, ranging from straightforward to more complex solutions. This situation is even more complex in cases of multiple network segments or data sharing across different security levels.

d) User management: Due to the absence of a common authentication and authorization mechanism, previously there has been no need for a centralized user management system. Instead, individual CSD vendors implemented their own solutions for user management. This encompassed users at client interfaces, at server-to-server interfaces and at internal interfaces. With the implementation of authentication and authorization within the coalition, a solution for a common user management system is now required. It is essential to ensure continued access to products from an external CSD while simultaneously verifying that such requests are legitimate. Therefore, it is necessary to distribute credentials effectively. An important aspect to consider is the maintainability of the system.

e) Compatibility with overall networks: As a consequence of the aforementioned need to be compatible with deployments within federated network of networks that may have overarching security requirements and offer their own security infrastructure, the specification must be as flexible as possible. In particular, instead of specifying a concrete mechanism, the standard should provide a solid guidance section including robust advice.

AUTHENTICATION AND AUTHORIZATION MECHANISMS

There are several well-established authentication and authorization mechanisms when it comes to HTTP(S)-based communication. In this section, we address some of the most common ones and evaluate their advantages and disadvantages for our use cases, especially focusing on the aspect of vulnerability.

a) Basic Auth: Basic Auth is an authentication scheme defined in RFC7617 (https://datatracker.ietf.org/doc/html/rfc7617). It involves sending the user's username and password in base64-encoded form in the *Authentication* header of every request, so the server can check these credentials directly and explicitly. As the name implies, this is a very simple approach that incurs security disadvantages. It should be noted that (if the connection is encrypted, i.e. using TLS) sending the credentials in plain text is not insecure in principle but can lead to problems related to SSL-termination at reverse proxies or similar components. A related problem is that even a single compromised request leaks the user's credentials (as opposed to a time-limited token in other methods). A further problem is the lack of any kind of built-in session and associated ability to log out and prevent further communication.

b) API Keys: API Keys are secrets that are sent in every request and are then correspondingly evaluated at the target server. They can be contained in either a header field or the URL. It is important to note that in contrast to Basic Auth this secret is not equal to the user's account password but is generally obtained via a website which is accessible by the user through a login. This makes the mechanism less vulnerable. An additional advantage is that API keys may be limited to a subset of APIs or only allow access to specific resources, e.g. having one API key per project. While in case of a compromised request the immediate consequences are similar (i.e., the attacker is fully able to make requests in place of the user) the API key can be more easily invalidated without input from the user. Because of this indirection API Key rotation schemes which automatically invalidate API keys after a certain duration are possible and common. In practice API keys are, however, only recommended for technical users (i.e. ones associated with an application) and not human users (Google Cloud, 2025). Due to many practical concerns with the theoretical advantages discussed above (Duncan, 2023), API keys inherit many of the security disadvantages of Basic Auth.

c) Certificate-Based Authentication: Certificate-based authentication relies on the client presenting a digital certificate when making requests to a server that binds the client's identity to a cryptographic key (Innocent Uzougbo

& Oyegbola Augustine, 2025). In case bi-directional verification is desired, Mutual TLS (mTLS) can be used where both client and server present their certificates. Similarly to API keys, certificate-based authentication is particularly suited for machine-to-machine communication where specific devices are authenticated using their assigned cryptographic key. Digital certificates may also be a component of more advanced techniques like Smartcard-based authentication methods (Lal et al., November 2016). If the underlying cryptographic methods are secure, certificate-based authentication provides a high level of security, however that comes at the cost of the large practical and organizational effort of establishing a Public Key Infrastructure (PKI) that supports the management of keys and Certificate Authorities (CAs).

d) OAuth2/OpenID Connect: Oauth2 (https://datatracker.ietf.org/doc/html/rfc6749) is an authorization framework to use tokens (transported in the *Authorization* header) to access services and resources. These tokens are only valid for a certain (usually quite small) amount of time and explicitly support so-called scopes to limit the permissions granted as much as possible. OpenID Connect extends OAuth2 to add authentication capabilities. OpenID Connect is both powerful and commonly used. However, its security relies on a robust and capable implementation that follows best practices and is properly configured. All of this is not easy due to the specification's inherent complexity (Innocent Uzougbo & Oyegbola Augustine, 2025).

Comparing Authentication Mechanisms

In principle, the actual security requirements for the authentication mechanism used depend on the network where the deployment is taking place. Coalition Shared Data is aiming to be used in an operational environment and network. To ensure interoperability between the different implementations and to verify operational process support CSDs are connected in experiments, exercises and trials.

For this purpose, Basic Auth is a good choice since it allows for authentication with no coordination effort outside of the exchange of credentials. Additionally, any mature software ecosystem will have support for Basic Auth on a technical level. The (possible) security disadvantages incurred are usually acceptable if only test data is used. It is, however, generally not acceptable for operational use due to security concerns (see above). Instead, one of the other options should be favored.

When it comes to API keys, an organizational disadvantage is that since API keys are strictly speaking not a predefined authentication scheme, the *Authentication* header field cannot be used and a custom header name like *X-API-KEY* must be coordinated between participants. The recommendation that they should only be used for machine-to-machine communication (i.e., technical users) can be accommodated by conceptually splitting the CSD-API into client-facing and server-facing endpoints and leveraging API keys only for the latter. Additionally, the more advanced management capabilities discussed above may be valuable in many deployments. Operational use seems feasible but generally only for

machine-to-machine communication. For exercise purposes, API keys are a solid alternative to Basic Auth for server-facing endpoints. For client-facing endpoints another mechanism must be used.

Using Certificate-Based authentication in a multinational environment would need an agreed-to supernational organization providing a PKI. Numerous technical details (including the concrete format of the certificates and certificate requests, their validity duration, and the revocation processes) must be thoroughly coordinated and communicated prior to any implementation. This is particularly challenging given the heterogeneous nature of the vendors and technologies involved. Consequently, while this method is usually not an option for exercises, it may be applicable for long-term, well-managed deployments in general, also for operational purposes.

In a scenario where a single entity controls the complete backend and can offer clients a stable, well-defined and uniform API, OpenID Connect is the de-facto standard way of handling authentication and authorization. However, in a decentralized, multinational environment it is difficult to use – all members would need a centralized identity provider (or a federation scheme). There is overhead in managing and coordinating the details needed to make this protocol work – from client names to the exact structure of the tokens, claim names etc. Similarly to API keys, OpenID Connect can support different kinds of flows for client-server (authentication code) and server-server communication (client credentials). From a security as well as from a tooling perspective, OpenID Connect is a good choice as an authentication mechanism – it overcomes many of the challenges of Basic Auth and API keys and is simpler to setup than certificate-based authentication. It can be used for both exercise and operational deployments.

IMPLEMENTATION EXPERIENCE

As discussed in Challenges of Authentication and Authorization there is no specified required common authentication mechanism in a CSD network. For this reason, it makes sense to support a wide array of mechanisms in the implementation of a CSD server. This chapter discusses our experiences implementing the mechanisms mentioned above.

When implementing standards of any kind but particularly security-related ones involving cryptography, it almost always makes sense to leverage existing technology – both for libraries and services. These are (in principle) publicly tested if not audited by experts, making bugs rare and in cases they do occur, they are quickly fixed. Thus, a mature ecosystem for the language used is important. An aspect of using popular services we experienced, which can be viewed both positively and negatively, is the ability to avoid coordination between partners by implicitly relying on particular behaviors of common services.

Generally, using commercials services for authentication and authorization is a reasonable option. However, in the CSD context, we consider a closed network that is not connected to the Internet. Therefore, only services that can be self-hosted are feasible. One such service is Keycloak (https://www.keycloak.org/) - a widely used open-source identity provider with support for multiple authentication methods and their versions (e.g.

different OAUTH flows) that can be self-hosted. In our environment Keycloak is already widely used. For this reason, we chose to use Keycloak as our identity provider when using OAUTH – for simpler authentication mechanisms like Basic Auth we relied on internal user management, allowing for maximum flexibility and (if necessary) code changes for workarounds. However, this requires self-management and storage of credentials. Depending on the threat model under consideration, it's associated accreditation procedure and the actual system characteristics this may be difficult. There are services that help with this like Hashicorp Vault (https://www.hashicorp.com/de/products/vault) which we leveraged in a prototype to store user credentials (both passwords and API keys) and may evaluate more in the future. With respect to certificate-based authentication we did not develop a particularly robust implementation due to a lack of support on a technical level.

Testing at Exercises

We tested our approach at CWIX (Coalition Warrior Interoperability Exercise) (NATO, 2025), an annual test event used to validate implementations against NATO standards. We successfully tested Basic Auth communication with other CSD vendors. However, there was no automatic mechanism for the distribution of user credentials. Instead, this had to be done manually.

For our internal communication (which may or may not use the same APIs as the multinational communication) a more heterogenous array of methods was used – specifically API Keys for service-to-service communication. If the credential distribution problem is solved there is potential to use them for communication with external servers too. We also performed tests on using OpenID Connect with a Keycloak identity provider to enable third-party front-end clients to communicate with our APIs. Due to the large number of moving parts when it comes to configuring Keycloak as well as the fast level of development and slight differences with each version informal discussions are difficult. We found that agreeing on a specific version and (ideally) providing an exact JSON export of the state of the identity provider (called a "realm export" in Keycloak) greatly speeds up the setup process. Due to the lack of test partners and the required effort certificate-based authentication was not tested.

Leading up to CWIX there are also risk reduction tests that take place in a cloud environment. In general, there exists the potential of experimenting with different auth mechanisms that may or may not be appropriate. This time those events were not used to do that but aided by the fact that code changes are much easier to incorporate than in an on-site exercise, this should be used to gather experience on this topic in the future.

CONCLUSION

Securing the interfaces of a multinational military standard with appropriate authentication and authorization mechanisms is not straightforward. We presented a selection of key challenges associated with this. Fundamentally, the necessary changes must be compatible with the underlying standard

STANAG 4559 or be made accordingly. Furthermore, there are organizational constraints. Coordination for unified configuration and user management among the participating coalition partners must be established. Lastly, it is important to note that data exchange via the CSD server typically occurs within a larger network comprising various systems and applications. Therefore, the chosen mechanism must be compatible with the respective deployment.

We elaborated on multiple authentication and authorization mechanisms, namely Basic Auth, API Keys, Certificate-based authentication, and OAuth2/OpenID Connect. Each of these mechanisms presents its own advantages and disadvantages. In principle, OAuth2/OpenID Connect is the most appropriate choice when considering the suitability in a multinational military environment and the trade-off between security and coordination effort. However, for compatibility with a variety of deployments, it is beneficial to be as versatile as possible and to support all variants. Our implementation experience has demonstrated that this is feasible.

Following the tests conducted, which confirmed authentication capability of the CSD server in general, part of the future work within the CST will involve drafting a guidance chapter on authentication and authorization. Additionally, we plan to expand the experience in our implementation with the different authorization and authentication mechanisms and perform further tests, possibly also in a cloud environment with other partners.

ACKNOWLEDGMENT

The standard STANAG 4559 is developed within the CST of STANAG 4559. The authors acknowledge valuable help and contributions from all partners within the CST.

REFERENCES

Duncan, A. (2023). Auth0 Blog. Available at: https://auth0.com/blog/why-migrate-from-api-keys-to-oauth2-access-tokens/ (Accessed: 11 July 2025).

Google Cloud, (2025). Google Cloud Documentation. Available at: https://cloud.google.com/endpoints/docs/openapi/when-why-api-key (Accessed: 11 July 2025).

Hagemann, L. & Klotz, P. (2024). From System High to Zero Trust: The Impact of Security Requirements on a Multinational Standard With Technical Specifications for Data Dissemination. International Conference on Human Systems Engineering and Design 2024, Volume 158.

Innocent Uzougbo, O. & Oyegbola Augustine, A. (2025). A Review of Authentication and Authorization Mechanisms in Zero Trust Architecture: Evolution and Efficiency. Tech-Sphere Journal of Pure and Applied Sciences, 2(1).

Lal, N. A., Prasad, S. & Farik, M., November (2016). A Review of Authentication Methods. International Journal Of Scientific & Technology Research, 5(11).

NATO Standardization Office (NSO) (2018a). STANAG 4559 - NATO Standard ISR Library Interfaces AND Services. Available at: https://nso.nato.int/nso/nsdd/main/standards/stanag-details/8838/EN (Accessed: 22 July 2025).

NATO Standardization Office (NSO) (2018b). NATO Standard ISR Library Interfaces-AEDP-17. Available at: https://nso.nato.int/nso/nsdd/main/standards/ap-details/2272/EN (Accessed: 22 July 2025).

NATO (2025). Coalition Warrior Interoperability Exercise. Available at: https://www.act.nato.int/activities/federated-interoperability/ (Accessed 23 July 2025).

Rodenbeck, R., Haferkorn, D. & Essendorfer, B. (2024). Interoperable Data Distribution Through Coalition Shared Data by Means of Standardization. International Conference on Human Systems Engineering and Design 2024, Volume 158.