Glahn, C. & Mazza, R. (2018). Integrating Native Mobile Apps into Institutional Educational-technology Ecosystems.
In D. Parsons, R. Power, A. Palalas, H. Hambrock & K. MacCallum (Eds.),
Proceedings of World Conference on Mobile and Contextual Learning 2018 (pp. 77-83).

Integrating Native Mobile Apps into Institutional Educational-technology Ecosystems

Christian Glahn
HTW Chur
Chur, Switzerland
christian.glahn@htwchur.ch

Riccardo Mazza
Università della Svizzera italiana
Lugano, Switzerland
riccardo.mazza@usi.ch

ABSTRACT
Identity management gains increasing relevance for mobile learning as mobile learning needs to be integrated into more complex learning scenarios. This paper discusses the different authorization strategies for mobile learning apps and an architecture for abstracting the relation between device-level and organisational trust domains. This architecture provides the foundation for authorization strategies that allow secure and transparent integration of native mobile learning apps into more complex learning environments. This paper discusses the implementation of a trust agent that creates a transparent layer that simplifies the user experience and reduces application complexity.

Author Keywords
Identity management, trust domains, authorization, authentication, mobile learning infrastructures

INTRODUCTION
Mobile learning has established its position in research and practice of technology enhanced learning (TEL). Higher education institutions invested in adopting mobile learning solutions to respond to their students' almost complete adoption of smart and mobile technologies such as smart phones and tablets. For scaling up mobile learning educational institutions need to address the challenges of integrating the related solutions into their overall educational infrastructure. The seamless integration of individual technologies into holistic learning experiences is one corner stone for building solutions for seamless learning (Wong & Looi, 2011, Wong, 2015). Therefore, mobile learning solutions need to integrate into the institutional technology-enhanced learning ecosystem.

In order to address the increasing relevance of smart mobile devices, many universities have created their own special purpose apps. Furthermore, there is a wide range of mobile apps offered from commercial providers. Many of these apps have potential applications in research and teaching but their integration with the universities' infrastructures is difficult or even impossible because the apps were not designed to integrate with on premise services or are tied to a specific infrastructure or system instance. Mobile apps that allow connections to different instances require an authentication protocol that are often cumbersome to configure on the mobile devices they run on. This can directly influence the adoption and scaling of mobile learning solutions due to customization cost and privacy concerns.

Integrating native mobile apps to the institutional infrastructure touches the interoperability between an app and the infrastructure it seeks to exchange data with. This has been identified as one of the key challenges within the field of contextual and mobile learning (Börner et al., 2010). Two interoperability related aspects are authentication and authorization. This aspect is typically ignored in discussions, data formats, and specifications on interoperability, but becomes important whenever the personal mobile learning experiences should be utilized for privileged access to institutional information, collaboration, student support, or assessment.

Connecting to institutional infrastructure is particularly a challenge for integrating native mobile apps because they come from a wide range of sources and run with elevated privileges on the users' mobile devices. This allows them to operate independently from a back-office infrastructure and offer more functional flexibility that progressive web-apps (PWA) have available. For example, native apps are not restricted to visual user interfaces and HTTP but have access to all communication channels exposed by the mobile devices' operating systems. This makes native apps the prime choice for personal information hubs in scenarios as discussed earlier (Glahn & Specht, 2010), which include connecting student devices to device ecosystems in smart classrooms and smart buildings. A major drawback of integrating native apps into institutional device ecosystems is that the institutions cannot verify the source of an app, easily. This poses a security risk because apps by rouge developers may gain access to personal data, as users have no indication whether any specific app is supported by their institutional infrastructure and infrastructure administrators cannot guide or restrict access of such apps.

This paper addresses three identity management challenges related to integrating mobile apps into educational infrastructures based on commonly found approaches for identifying users: infrastructure discovery, authentication, and app authorization. It analyses three scenarios for granting mobile apps authorized access to institutional infrastructures and relates these scenarios to a trusted infrastructure architecture that incorporates mobile apps. On the ground of this architecture and applying contemporary interoperability specifications, this paper addresses the user experience for increasing the secured access of institutional infrastructures by mobile apps as an attempt to leverage the potential of mobile apps in higher education.

BACKGROUND

Mobile learning apps allow to mediate in situ personal learning experiences. By being installed on the learners' personal handheld devices these apps enable more immediate, ubiquitous, and contextualized access to and collection of information, communication, or collaboration (Sheehy, Ferguson, & Clough, 2014). As long as a mobile app accesses public information and does not need to exchange information with other devices, anonymous use of such apps is completely sufficient. However, anonymous use of mobile apps is insufficient if mobile apps are part of more complex learning or educational scenarios that involve access to privileged or personal information, personalized support, or communication between learners. Especially formal education demands identifying the individual actors, their performances, and their roles whenever the related learning activities are part of assessment and grading processes. Therefore, mobile learning apps need authentication and authorization if they integrated into larger learning experiences and organizational processes. Secure and reliable authentication and authorization services for mobile apps also become increasingly relevant as the legal regulations for data protection and privacy evolve (European Parliament and the Commission, 2016).

The ADL Total Learning Architecture (TLA) is a new attempt for operationalizing the interoperability of educational technologies. It follows the general design principle of SCORM (ADL Initiative, 2004) of integrating interoperability standards and specifications into a reference framework for implementing and delivering TEL solutions. Different to SCORM this new reference framework is not restricted to web-based learning but seeks to provide guidelines for a wide range of TEL appliances, including games, simulators, mobile learning apps, as well as smart environments. This aims at reducing costs for integrating, maintaining, and scaling complex TEL appliances.

The ADL TLA identifies four central components that are shared by all educational systems:

1. learning resources,
2. experience tracking,
3. competence monitoring for process orchestration and assessment, and
4. identity and preference management.

The identity and preference management component identifies individual actors in the environment and provides their preferences and needs to other components of a learning environment. This component also enables data persistency and process consistency for the other components of the architecture, such as role and access management, activity enrollment, as well as for personalization. The identity and preference management component has four key functions:

1. authentication,
2. authorization,
3. identification, and
4. process independent preferences

While educational approaches, take these elements for granted, any technical has to provide these functions in order to differentiate individual actors. This particularly important for recognizing learning and formal education. The ADL TLA does not identity and preference management as a new component for educational technologies. Instead this component has to comply to existing interoperability standards for these functions.

Standardized authentication and identity services are not new in the educational realm: RADIUS (Aboba, Calhoun, 2003) is widely used for distributed authorized access of academic network infrastructures, most notably in the international eduroam network (GEANT, 2016). The SAML2 protocol (Cantor, Kemp, Philpott, & Maler, 2009) offers interoperable authorization and identity functions for interactive web-services such as virtual learning environments. The features of SAML2 include distributed authorization within so called service federations. This allows the integration of local identity services and shared resources into trust domains. These trust domains have been implemented for accessing IT services and resources in regional, national, and international academic networks.

SAML2 relies on XML transactions over HTTP in a web-browser environment. The specifics of the protocol are complex to implement and make it hard to extent for new application scenarios. These limitations have led to the development of OAuth (Hard, 2012). OAuth defines a framework that utilizes network protocol features instead of building an alternate protocol on top of the network protocol. While OAuth shares many concepts with the older SAML2 protocol, it reduces complexity and adds more flexibility for different authorization scenarios. Another OAuth feature is its modular structure, which allows extensions into order to meet the requirements of emerging application scenarios and use cases. This modular structure allows applications to limit the supported OAuth features to a necessary minimum, while retaining interoperability with more comprehensive solutions. Finally, OAuth relies on the JSON data format (Bray, 2017), which simplifies the integration of OAuth into modern applications even further. While SAML2 incorporates identity management features, these are deliberately absent in OAuth. Identity management has to get added to OAuth via the OpenID Connect (OIDC) specification (Sakimura, et al., 2014). OIDC utilizes OAuth's modular structure for integrating identity management features to the framework.

One important extension of OAuth is its flexibility towards the runtime environment as it makes no assumptions about runtime capabilities. This allows to support OAuth even for native apps on mobile devices (Denniss & Bradley, 2018) and

holds the potential to link to apps, appliances, and devices without visual user interfaces, such as wearables, securely into complex scenarios and experiences (Kolamunna et al. 2017).

The present efforts of providing authorization and identity services to native mobile apps rely on the assumption that the users' authorizing party is known to a native app prior to the authorization (Denniss & Bradley, 2018). This is achieved by configuring valid authorization parties directly into the app and registering it with them; or by asking users for the URL of the authorizing party. Configuring authorizing parties directly into an app is only useful for cases with a very limited number of authorizing parties. Asking URLs from users is only useful for native apps in desktop environments. On mobile devices this approach is typically cumbersome and error prone. This approach cannot be justified if more than one service should be connected to an app. In both cases rouge developers can gain knowledge about the authorizing parties of their users even when no access is granted.

RESEARCH QUESTION

The present research targets the development of a secure and user-friendly solution for granting native apps access to institutional and shared IT infrastructures. The present research addresses three questions for research:

1. What approaches to identity and preference management can be found for mobile apps used in higher education?
2. What use-cases are relevant for different types of mobile apps?
3. Is it possible to avoid the exposure of identity related information prior to a successful authorization?

The first question targets the status quo of handling the user identity of mobile apps. This question targets also the types of mobile learning solutions.

The second question categorizes the identity related use cases for mobile apps. This question answers to the functional requirements of the targeted solution.

The third and final question focuses on the limitations of the current best practices. While it is primarily security-related, it also considers the user experience during the authorization.

CONTEXT

These questions were approached in the context of Swiss higher education sector. Swiss higher education offers TEL and other services to a national federation called SWITCHaai. This federation allows to access local and shared services as well as services offered by other institutions using the user name and password of one's home institution. The SWITCHaai federation is used frequently by more than 250,000 distinct users in 65 institutions. The federation offers access to more than 600 service instances, of which library and TEL services such as virtual learning environments and video conferencing are used, most intensively. SWITCHaai is part of the eduGAIN network of European academic federations.

Over the last 8 years the participating institutions developed several mobile solutions for mostly educational purposes. Developers and practitioners from academic institutions exchange their status, progress and challenges in the eduhub SIG Mobile Learning, a national community of practice on mobile learning in higher education. The apps available in the institutions range from classroom response systems via communication and self-study apps to location-based inquiry apps.

TRUST DOMAINS OF EDUCATIONAL INFRASTRUCTURES

The federated identity management of the national academic network in Switzerland traditionally assumed only services that are accessed via the web-browser. Through responsive web-apps these services are also available on mobile devices. The traditional trust domain comprises of a network of identity providers (IDP) and a network of resource providers (RP). IDPs can authenticate users and can forward identity related information to RP. IDPs are connected to the user management on the different institutions. RPs, on the other hand, offer resource services to their users. Within the federation four types of services can be distinguished.

1. Restricted academic services can only get accessed by members of selected institutions. These services are commonly restricted to the members of the hosting institution.
2. Open academic services allow all federation members to access a service of one institution.
3. Shared academic services are very similar to open academic services with the difference that these services are part of the national academic infrastructure.
4. Licensed services are services offered by commercial parties. All federation members can access these services, but access to individual functions or resources might be restricted by bilateral license agreements.

The trust domain is traditionally limited to the network of IDPs and RPs. Members access the RP via their Web-browser through TLS encrypted sessions, indifferently if they access these services from a desktop computer or from a mobile device.

If native apps are added, we typically find two relevant trust domains: firstly, the federation's trust domain of the academic federation with IDPs and RPs; secondly, the app-store trust domain that comprises the apps that were authorized on the app store of the mobile operating system providers. The two trust domains overlap for apps that are tied to an RP or to an IDP. The trust connection between a federation and a mobile app is typically defined by the identity of the app provider and the RP/IDP provider.

OAuth's native app authorization specification (Denniss & Bradley, 2018) requires native apps to authorize themselves against an IDP to link into the federation's trust domain by the use of the device's web-browser. In this scenario the native app is known to an IDP from a prior registration.

An alternative approach has been previously discussed (Madsen, Jain, Zmolek, & Bradley, 2015) but has not been finalized for standardization. This approach depends on a special app, a so-called trust agent, that is linked to one or more IDP services or "authorization parties". It shares a federation's trust domain via the identity of the IDP provider and the app provider. In this scenario, the trust agent app bridges between the federation and the app-store trust domains and acts as a gatekeeper for other native apps.

The two approaches for authenticating native apps inform on the general alignment of the trust domains: native apps require an additional component on the mobile device that links the federation and the devices trust domains. This component must be external to the native app in order to ensure the integrity of the authorization process. In figure 1 this alignment is visualized as the "trust domain link" component, which can be a native app in the style of a trust agent or web-browser based as specified for OAuth's native app authorization specification (Denniss & Bradley, 2018).

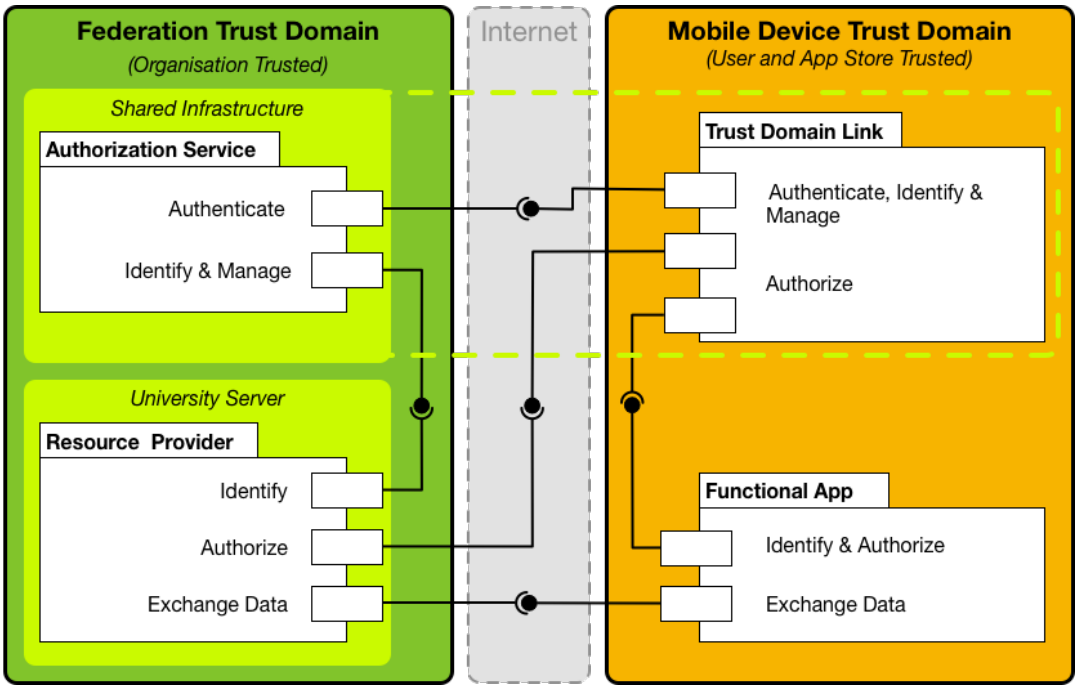

Figure 1: Outline of the relations and components in the different trust domains.

APPROACHES TO APP AUTHENTICATION

While the majority of mobile solutions in the Swiss academic federation are web-applications, several different strategies for authorizing native mobile apps were identified while analyzing 15 native apps connect to RP services in the national federation in the process of answering the first research question of this paper.

The "anonymous" strategy indicates that the users need not to identify or authenticate themselves when using the app. This strategy can be found for apps that provide mobile access to public resources as well as in classroom response systems.

The "pseudonymization" strategy asks the users to select a unique pseudonym for a session, which device users identify themselves through a self-chosen nickname. We found this strategy in apps for video conferencing and chat services.

The "IP restriction" strategy restricts the access of an app to an RP based on the IP of the user's device. This strategy can be found mostly for restricted academic services.

The "geo-location" strategy is similar to the IP restriction strategy but relies on the geographical position of a device. Combined with pseudonymization this strategy has been found in a classroom response solution.

The "user token" strategy relies on the users manually generate user tokens in the web-based user interface and then copy them into the app. This strategy has been found in LMS and chat apps.

SAML2 authentication relies on an app-specific web-based authentication component. After successful authorization the web-component forwards the SAML2 credentials to the respective app. This has been found for an LMS app and a commercial content provider app.

The "proprietary authentication" strategy uses an alternative user authentication for granting an app access to a service. This strategy has been only found with apps for restricted academic services.

Custom OAuth2 protocol has been found for a file-management and -sharing app. The users authenticate in the browser using an OAuth2 flow on the web and are then forwarded into a native app via a custom URL.

The strategies "anonymous", "pseudonymization", "IP restriction", as well as "geo-location" do not rely on any form of authentication as they simply guide access to a resource. Therefore, these strategies are considered as insecure approaches to access privacy related data.

The strategies "user token" and SAML2 authentication integrate well with the SAML2 federation. The "user token" is cumbersome to handle, because the users have to first generate the token and then copy it to its device. The SAML2 strategy is easier to handle but has no build-in mechanisms to mitigate known attacks through custom protocol re-registration on mobile devices (Denniss & Bradley, 2018).

The "proprietary authentication" as well as the "custom OAuth2 authentication" strategies are, both, alternate approaches to identity management and typically rely on different credentials than for authenticating with the federation.

SCENARIOS FOR CONNECTING MOBILE APPS TO INSTITUTIONAL INFRASTRUCTURES

From the analysis of the solutions of the mobile learning solutions we identified three relevant scenarios for authorized access of mobile apps with the federated infrastructure.

The first scenario are progressive and responsive web-apps that operate entirely within the devices' web-browser typically have a backend service that can be registered to the federation as a regular RP service. These apps can operate completely within the conventional trust domain of connected services.

The second scenario are native apps that operate entirely as a frontend for a single service in the federation trust domain. This allows the app and the service to be tightly coupled. Consequently, the app "knows" the service it needs to connect to and vice-versa. This creates a mini trust domain including the app and the service. From the perspective of the federation this mini trust domain is indistinguishable from the service itself: An app may directly authenticate with an IDP in the federation or use its service's OAuth interfaces, if available. This scenario also covers settings where native apps need a cloud-based persistency layer for synchronizing user data across devices. In this special case, only the app is exposed to the federation, while its backend services trust the authorization codes provided by the app.

The third scenario covers native apps that rely on interoperable protocols and that can connect and orchestrate several RP services, simultaneously. In this scenario, the RP services and a native app are loosely coupled and do not form a mini trust domain but are independent entities, both, from the perspective of the federation as well as from the user perspective. In order to access any service in the federation it is necessary that the services are exposed to the app, while an IDP should only grant authorizations to services that support the protocols supported by the app. This scenario can be considered as the general case of the second scenario. However, for users in large federation trust domains exposing individual services by the means of the second scenario is difficult, time consuming, and error prone. Obtaining the different authorizations also adds logical complexity to the app that is unrelated to its function.

ENSURING INTEGRITY ACROSS TRUST DOMAINS

In the context of the third research question of this paper, we revisit the concepts of trust agents for loosely coupled native apps within the third scenario. A trust agent is an app that runs on the users' devices and is authenticated for a user with the respective IDP services of the federation. The special positioning of the trust agent within the federation enables it to identify the various the RP services in its trust domain.

If a native app or an appliance seeks to access a privileged information for the device user, the app initiates a request to the devices' operating system. In this request the native app identifies itself and requests access to services that support a given set of protocols. At this point the native app does not know which services or trust domains it will be authorized for. Instead, the user is requested to select an appropriate trust agent for the app. The trust agent filters the federation's RP services that meet the protocol authorization request and requests confirmation by the user.

In order to increase the control of the users on which services they wish to use, they have the opportunity to select those RP services for authorization from the list of matching RP services. The users can further limit the list of available services of the trust agent to those services that are offered by the institution or to those that were previously accessed. This allows the users to make an informed decision which services they want to expose to a native app and to grant access to multiple services in a single step. The user experience is illustrated in figure 2.

Because only the trust agent needs to authenticate the device's user to the federation trust domain, the exposure of user credentials is further limited. Once a trust agent has established a user session, it exchanges a cryptographic key for this session with the IDP service. This proof-of-possession (Jones, Bradley, & Tschofenig, 2016) ensures the integrity of following requests tied to the user.

Once the user consented to the authorization of the selected RP services, the trust agent initiates a series of authorization requests to these services using the normal authorization protocol as defined by OAuth. This enables the RP services and

the IDP service to verify the request and impose additional measures for the requesting app, including the rejection of access to a specific service. This allows to enforce privacy and security policies at the level of the federation as well as at the level of individual RP services. The requesting app will only receive information about data endpoints and authorization tokens for services with successful authorization.

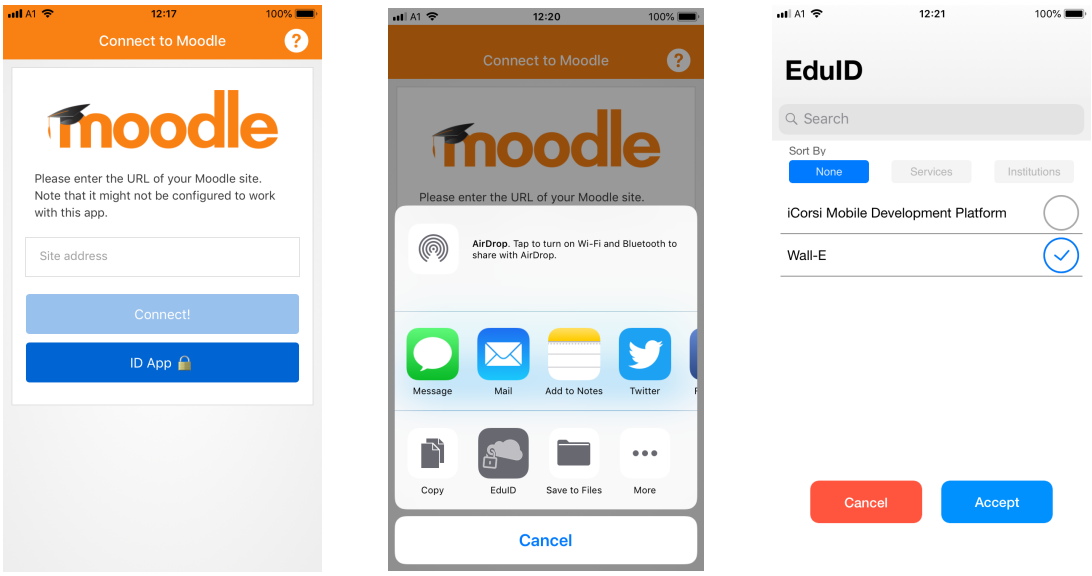

Figure 2: Illustration of the User Experience of a trust agent authorization.

In case of failed authorizations, the trust agent displays any errors to the user and does not forward the related information to the requesting app. In case of unsuccessful authorizations, the requesting app will receive no information about the federation or the reason for the rejection, while the trust agent can make these steps fully transparent to the user.

For app developers the concept of the trust agent removes the need to implement or integrate a full authorization stack in order to access privileged data from RP services in a federation trust domain. Instead application developers need only to identify an application and its protocol needs. This reduces the complexity of the application logic and releases the app developer to provide and maintain customized solutions for different federations.

The integrity of the authorizations is ensured by cryptographically signed requests between all involved components using public key cryptography using JSON Web-Tokens (Jones, Bradley, & Sakimura, 2015). This limits the potential for forging valid requests for a party as the private keys that are used for signing requests are not exposed and can be securely stored. At the same time all involved parties can verify the validity of a signature through shared public keys. As all authorizations are independent from each other, the integrity of the authorizations is ensured by independent authorizations that are each tied to a tuple of a RP service, a native app, a mobile device, and a federation user. This results in restricted authorizations for individual apps on the users' devices to access the user's RP services. In case of security or privacy related incidents, this allows device users as well as system administrators and federation managers to revoke privileged access for apps or to RP services for selected devices and user audiences.

The complexity of the technical authorization process is hidden from user. Through the concept of the trust agent it is possible to improve the security related user experience independently from the native apps that offer functional experiences in a learning environment. These concepts were implemented and tested in a functional proof of concept for Android and iOS devices and a modified version of the Moodle App.

CONCLUSIONS

This paper presents a concept of trust agents to create transparent and secure access to complex service federations for native mobile apps. It identified different practices for exchanging personal data with services and other users. While not all of these practices are appropriate for the integration of native apps into the complex learning scenarios of institutionalized learning and education, technical standards exist that support the integrity and transparency of data access with user-friendly and easy to understand authorizations. This is not only relevant for satisfying the increasing legal demands for data protection and privacy, but also for creating more complex and seamless learning experiences in the evolving device-ecosystems, in which increasingly often devices are used without visual user interfaces. The concept of the trust agent offers a generalization for native app authentication that reduces complexity of user experiences and application logic.

ACKNOWLEDGMENTS

The research and development underpinning this paper has been partially supported by the swissuniversities Scientific Information Programme.

REFERENCES

Aboba, B. & Calhoun, P. (2003). RFC 5080: RADIUS (Remote Authentication Dial In User Service), Support For Extensible Authentication Protocol (EAP) Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc5080

ADL Initiative (no date) Total Learning Architecture. https://www.adlnet.gov/projects/tla

ADL Initiative (2004) SCORM 1.1, Technical Specification. http://adlnet.gov/research/SCORM/SCORM-2004-4th-edition/

Börner, D., Glahn, C., Stoyanov, S., Kalz, M., & Specht, M. (2010). Expert concept mapping study on mobile learning. Campus-Wide Information Systems, 27(4), 240-253.

Bray, T. (ed.) (2017). RFC 8259: The JavaScript Object Notation (JSON) Data Interchange Format. Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc8259

Cantor, S., Kemp, J., Philpott, R., & Maler, E. (eds) (2009) Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. https://www.oasis-open.org/committees/download.php/35711/sstc-saml-core-errata-2.0-wd-06-diff.pdf

Denniss, W., & Bradley, J. (eds.) (2018). RFC 8252: OAuth 2.0 for Native Apps. Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc8252

European Parliament and the Council (2016) Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.-ENG&toc=OJ:L:2016:119:TOC

Glahn, C., & Specht, M. (2010). Embedding Moodle into Ubiquitous Computing Environments. In M. Montebello, et al. (Eds.), 9th World Conference on Mobile and Contextual Learning (MLearn2010; pp. 100-107). October, 19-22, 2010, Valletta, Malta.

Hard, D. (ed). (2012) RFC 6749: The OAuth 2.0 Authorization Framework. Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc6749

Jones, M., Bradley, J. & Sakimura, N. (eds) (2015). RFC 7519: JSON Web Token (JWT). Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc7519

Jones, M., Bradley, J. & Tschofenig, H. (eds) (2016). RFC 7800: Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs). Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc7800

Kolamunna, H., Chauhan, J., Hu, Y., Thilakarathna, K., Perino, D., Makaroff, D., Seneviratne, A. (2017). Are Wearables Ready for Secure and Direct Internet Communication? GetMobile: Mobile Comp. and Comm., 21(3), 5-10. doi:10.1145/3161587.3161589

Madsen, P., Jain, A., Zmolek, A., & Bradley, T. (2015). OpenID Connect Native Application Token Agent Core 1.0 (draft). OpenID Foundation. https://openid.bitbucket.io/draft-native-application-agent-core-01.html

Sakimura, N., Jones, M., Bradley, J., de Medeiros, B. & Mortimore, C. (eds) (2014). OpenID Connect Core 1.0. OpenID Foundation. http://openid.net/specs/openid-connect-core-1_0.html

Sheehy, K., Ferguson, R., & Clough, G. (2014). Augmented education, bringing real and virtual learning together. New Your: Plagrave Macmillan.

Wong, L.-H. (2015). A Brief History of Mobile Seamless Learning. In L.-H. Wong, M. Milrad, & M. Specht (Eds.) *Seamless Learning in the Age of Mobile Connectivity* (pp. 3–40). Singapore: Springer Singapore.

Wong, L.-H., & Looi, C.-K. (2011). What seams do we remove in mobile-assisted seamless learning? A critical review of the literature. Computers & Education, 57(4), 2364–2381.