Reliable e-Assessment with GIT – Practical Considerations and Implementation

Bastian Küppers[1], Marius Politze[2], Ulrik Schroeder[3]

[1]IT Center, RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, kueppers@itc.rwth-aachen.de
[2]IT Center, RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, politze@itc.rwth-aachen.de
[3]Learning Technologies Research Group, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, schroeder@informatik.rwth-aachen.de

Keywords
Computer based examinations, Computer aided examinations, e-Assessment, e-Learning, BYOD, GIT, SOA

1. ABSTRACT

The introduction of e-Assessment is hindered by reservations, despite e-Assessment being a valuable tool for teaching and examining in general. These reservations concern mainly the fairness and reliability of e-Assessment as well as financial questions. The latter can be solved in practice by utilizing BYOD, since most of today's students already possess devices that are capable of being used for e-Assessment. This poses, however, additional challenges to the fairness and reliability of e-Assessment. This paper presents an approach that utilizes cryptographic methods, like public key cryptography and digital signatures, and the version control system GIT in order to overcome this challenges. Based on these techniques a framework for e-Assessment was implemented as client-server architecture.

2. MOTIVATION

In accordance with the general trend in society, teaching at institutes of higher education in Germany is increasingly digitized (Hochschulforum Digitalisierung, 2016). This development has different manifestations, for example the usage of learning management systems (Hochschulforum Digitalisierung, 2016) or mobile apps (Politze, Schaffert, & Decker, 2016). The incorporation of digital elements, however, is not limited to lectures, but also tutorials keep up with this trend. In programming exercises, for example, it is rather common that the students can work on the assignments with their own devices. Despite e-Assessment being already a part of lectures in form of self-tests and formative assessment, it has not yet been established for examinations (Hochschulforum Digitalisierung, 2015). The same situation has also been reported for other countries, like the United Kingdom (Walker & Handley, 2016), Greece (Terzis & Economides, 2011), the United States of America (Luecht & Sireci, 2011) and Australia (Birch & Burnett, 2009). Hence, there is a media disruption between lectures and tutorials, which are digitized, and examinations, which are in general not digitized. Figure 1 illustrates this situation.

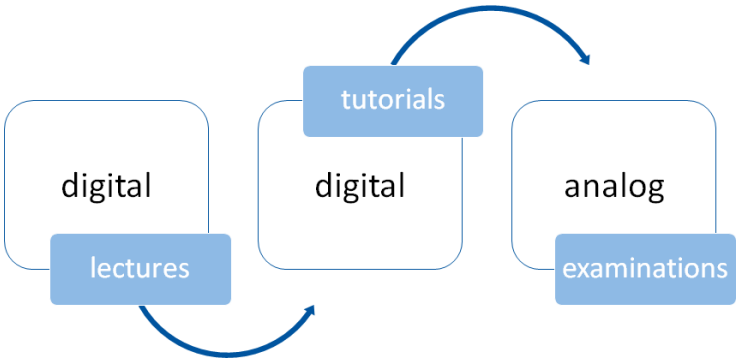

Figure 1: Current state of the digitalization in university courses.

Staying with examinations on paper is mainly caused by two reasons: Reservations against e-Assessment and money. The reservations concern mostly the fairness and reliability of digital examination systems (Vogt & Schneider, 2009). Reservations hinder the introduction of e-Assessment quite drastically, since acceptance, especially by the students, is one important key factor for the success of e-Assessment (Fluck, Pullen, & Harper, 2009) (Terzis & Economides, 2011). Beyond reservations against e-Assessment, the costs for maintaining a centrally managed IT infrastructure for e-Assessment hinder the introduction of e-Assessment additionally, since not every institution can afford the money for the hardware and the administrative effort. That those costs are quite high is reported by several institutes of higher education, which maintain a centrally managed IT infrastructure for e-Assessment, for example the University of Duisburg-Essen (Biella, Engert, & Huth, 2009) and the University of Bremen (Bücking, 2010).

E-Assessment, however, offers also significant benefits, making it worth considering. Especially in computer science education, the introduction of e-Assessment provides remarkably benefits. First and foremost, e-Assessments bring the main subject of study, namely the computer, into the examination, but there are also other facets, which improve the setting of the examinations for both, students and lecturers.

Often, tutorials are held in addition to the lectures to provide hands-on experience besides theoretical education. These tutorials introduce domain specific tools to the students. In a programming course, the students normally get used to integrated development environments, e.g. Eclipse or NetBeans, during the tutorials, but are currently most often asked to write the examinations on paper. Therefore, these tools can also be used in the examinations, closing the previously described gap between tutorial and examination.

The same tools that the students can use throughout the examinations, are also available to be used by the correctors, reducing the effort for correction considerably (Jara & Molina Madrid, 2015) (Vogt & Schneider, 2009). Parts of the examination can even be corrected semi-automatically. In programming courses, for example, a set of unit tests can be used to determine whether a student's code fulfils all the requirements demanded by the examination. Only if some of these tests fail, the corrector has to have a deeper look into the student's exam. Additionally, also the effort for correcting the other parts of the examination is lowered, because the readability is clearly improved in comparison to handwritten examinations.

By providing domain-specific tools, also the proficiency level of the examinations can be increased remarkably. Considering Krathwohl's revised version of Bloom's Taxonomy of Educational Objectives (Krathwohl, 2002), assessing the more complex levels of the taxonomy, like Evaluate and Create, can be achieved in a more realistic fashion, because the domain-specific tools can take care of the lower levels of the taxonomy, so that the students can focus on the higher levels. In programming courses, the integrated development environment for example provides auto completion; therefore, the students do not need to remember every keyword of a programming language.

3. GOALS

The main goal of the previously mentioned framework for e-Assessment is to provide a solution for common issues related to e-Assessment. Therefore, this framework could potentially boost the implementation of e-Assessment at institutes of higher education. In order to possibly achieve this main goal, two sub goals have been identified that have to be fulfilled. First, the reservations against e-Assessment have to be eliminated and second a solution for financial issues has to be found.

Since most students nowadays already possess own devices that are suitable for e-Assessment (Dahlstrom, Brooks, Grajek, & Reeves, 2015) (Poll, 2015) (Willige, 2016), Bring Your Own Device (BYOD) is a potential solution to the financial issues. By implementing BYOD, the institutes of higher education would not have to maintain a full-blown IT infrastructure including servers and workstations. Instead, maintaining a server infrastructure would be sufficient, because the students would bring their own workstations. Therefore, we chose to make BYOD a key element of the e-Assessment framework.

When considering BYOD as key element of the framework, the fairness and reliability of e-Assessment get even more important, because the students' devices are not per se under the control of the examiners. Therefore, we consider it very important that the framework provides a

comprehensible reliability in order to be able to reduce reservations against e-Assessment. Hence, we plan to release the framework under an open source license, once the research project is finished. That enables everyone to examine the software in detail and therefore helps to build trust, as studies suggest (Boulanger, 2005) (Miller, et al., 1995). Furthermore, it allows building a community that helps to maintain and improve the software, for example by reporting bugs or design flaws. For the same reason, we decided to use the version control system GIT as storage backend for storing the student's results: it is open source and well tested and maintained. Additionally, GIT provides a versioning of the submitted data and therefore enables an examiner to reconstruct the steps taken by a student while solving an assignment.

Beyond making the framework open source in order to build trust, the data that is processed during an examination has to be handle in a trustworthy way to further support the acceptance of e-Assessment. That means especially that the students' results have to be stored safely. Hence, it must be ensured that the results cannot be manipulated after the students have handed in and, additionally, each set of results has to be clearly relatable to a particular student (Dahinden, 2014). The usage of BYOD allows us to adapt the concept presented by Dahinden, because the students work on their own devices, i.e. on a *trusted platform*. Therefore, we can utilize cryptographic methods that do not have to rely on temporary key pairs generated for an examination, but can work with key pairs that are certified by a certificate authority, for example the DFN-PKI (DFN-PKI, 2016). That would not be possible on workstations in a PC pool, because these computers are not under the control of the students. Therefore, these systems can be regarded as *untrusted platforms* from the students' perspective.

Another issue that has to be dealt with, especially in a BYOD context, is the equality of treatment, i.e. that every student has to have the same chances of performing well in the examination as every other student. Besides being an ethical guideline that should be obeyed by examiners, it can be mandatory by law. In Germany, for example, an equality of treatment is enforced by Article 3 of the Basic Law for the Federal Republic of Germany. In the worst case, the students' devices are all different from each other, therefore it is virtually impossible to have an equality of treatment in terms of hardware. It is, however, possible to design the e-Assessment framework in a way that deals with this issue. Hence, we based the design of the e-Assessment framework on a client-server architecture. For the assessment client, which will be executed on the students' devices, we chose to implement it as a light-weight web application. Therefore, every device that is capable of running a web browser can be used throughout an examination. Any device whose processing capabilities exceed these requirements does not have an advantage.

4. RELATED RESEARCH

E-Assessment at institutes of higher education is clearly not a new phenomenon, despite still a recent one. There are several ways how e-Assessment is carried out at the moment. These spread from the usage of learning management systems (Michel, Goertz, Radomski, Fritsch, & Baschour, 2015) and their built-in possibilities, like multiple-choice questions or free text assignments, up to the utilization of special software, for example the OPS software at RWTH Aachen University (Janßen, et al., 2014).

And also BYOD for examinations at institutes of higher education is not a new phenomenon. It is already used at several institutes of higher education around the globe (Küppers & Schroeder, 2016).

In their article Brauckmann et al. describe how certificate authorities (CAs) try to standardize documentation of certificate policies. They show the importance of standardized policies. Certificate policies are mostly documents of 90 pages that, among other things, describe how security of certificates is organized and how certificate requestors authenticate. They furthermore note that common CAs formulate their policies quite differently and users should review them individually before basing processes on their certificates. (Brauckmann & Gröper, 2013).

Ubiquitous access and programmable interfaces (APIs) are becoming more important as the number of smart devices and data intensive applications rises: Not only in context of e-learning but also as parts of every users' daily routines. The Horizon report, one of the most regarded studies concerning the development of education lists Mobile Learning, Adaptive Learning Technologies and Next Generation Learning Management Systems among the important developments currently faced in higher education (Adams Becker, et al., 2017). To meet the future challenges universities approach

interfaces, allowing integration of existing systems into new applications for students and employees. The works of Mincer-Daszkiewicz and Barata et al. show two practical examples of APIs in the field of university administration (Mincer-Daszkiewicz, 2014) (Barata, Silva, Martinho, Cruz, & Guerra e Silva, 2014). The process oriented architecture presented by Politze et al. generalizes for several personalized eLearning applications but aims to cover more university processes (Politze, Schaffert, & Decker, 2016).

In terms of software development, recent process supporting applications focus on small building blocks that feature integration and loose coupling. Micro service architectures as proposed by Namiot et al. reduce dependencies in the software development process often found in former monolithic applications (Namiot & Sneps-Sneppe, 2014). Virtualization allows separating these micro services from one another on the same hardware platforms. However Schleicher et al. show that virtualization and cloud environments pose additional dependencies and different requirements to the software development process and the deployed software (Schleicher, Vogler, Inzinger, & Dustdar, 2015).

5. IMPLEMENTATION

For the implementation of the assessment client we utilized the electron framework (GitHub, 2017), which itself is based on the NodeJS framework (Node.js Foundation, 2017). Thus, the client fulfills our design goal to be a light-weight web application, because the needed runtime environment can be considered equivalent to a web browser. Furthermore, the electron framework provides support for the three major platforms for mobile computers, Windows, Linux and MacOS. We plan to support also ChromeOS, Android and iOS eventually, which was another reason for the implementation of the client as web application. Therefore, no group of students has a disadvantage by having a not-supported operating system installed on their devices. Figure 2 shows a screenshot of the assessment client, which has loaded a programming assignment.

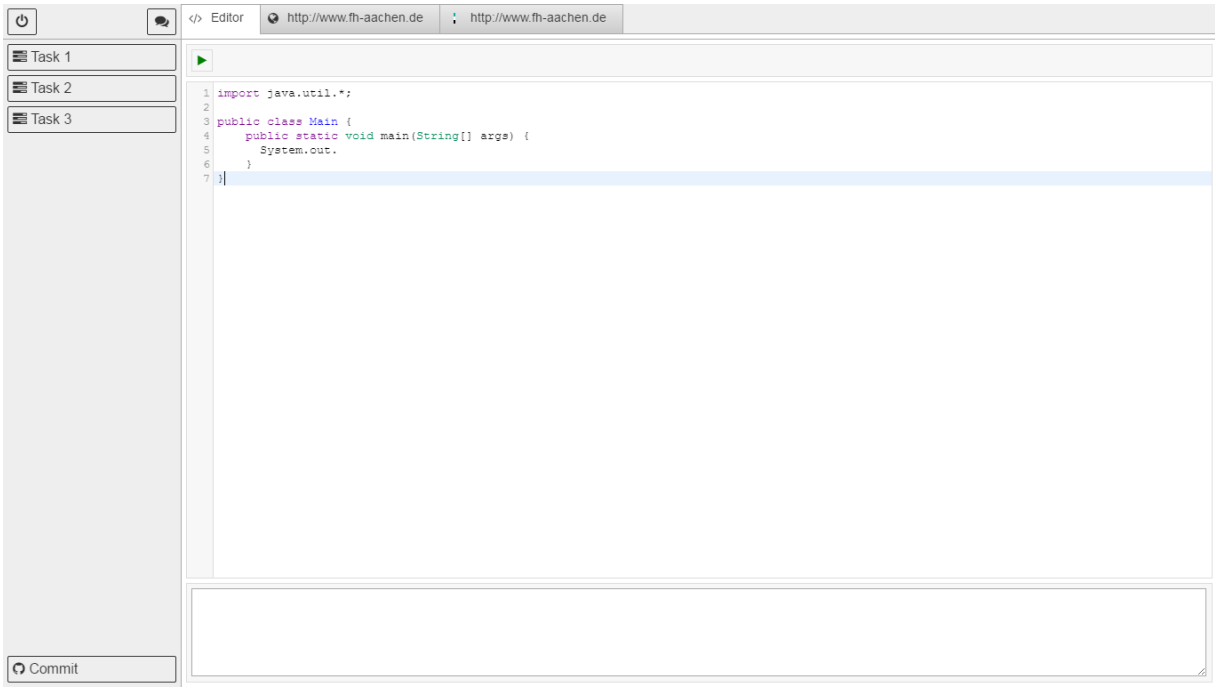

Figure 2: Screenshot of the assessment client.

The implementation is designed in a modular fashion (see Figure 3), which allows to easily extend the framework later on. Therefore, it is also possible to use another tool than GIT as storage backend for saving the students' results. Additionally, it is possible to introduce new types of assignments or to extend existing ones.

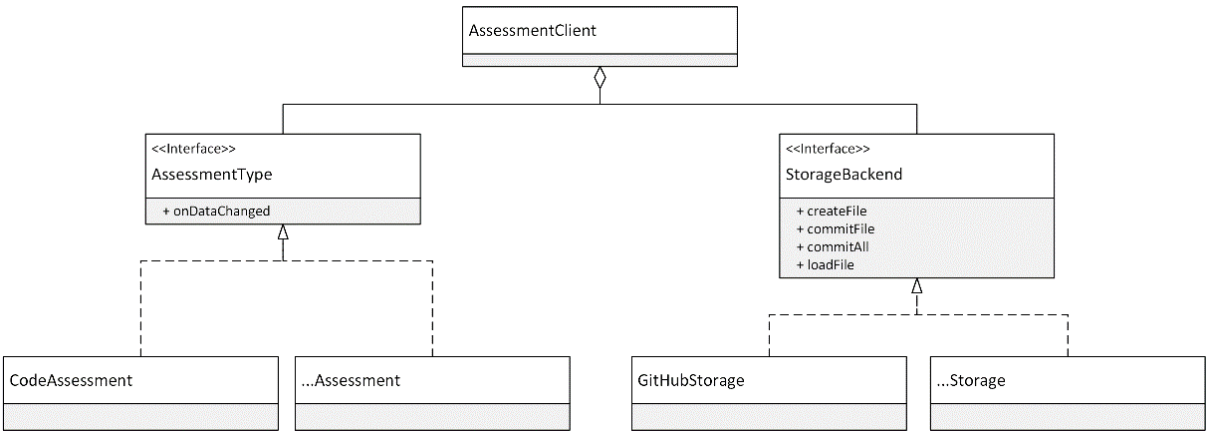

Figure 3: Modular architecture of the assessment client.

For a proof-of-concept we decided to use GitHub (GitHub, 2017) as storage backend. Therefore, we could use existing infrastructure and concentrate on the implementation of the prototype. To interact with the GitHub server, we a publicly available API that GitHub provides. This API is based on REST (representational state transfer), which is a programming paradigm for distributed systems, for example webservices (Fielding & Taylor, 2002). Due to the modular architecture, as already mentioned, it would be possible to use other services as storage backend, for example local instances of GitLab (GitLab Inc., 2017) at the institute of higher education.

In order to make the students' results relatable to a particular student, each change on the storage backend is signed with the student's private key. In case of GitHub, that means that every commit is signed. The built-in crypto modules of NodeJS were used to compute the signatures.

When the assessment client starts, it loads a configuration from the assessment server, which contains all information about the assignments. Each assignment is related to a specific file on the storage backend. If that file already exists on the storage backend, the assessment client loads that file and makes the contents available to the student. That enables the examiner to provide the students with a prepared skeleton for an assignment. If the file does not exist on the storage backend, an empty file is created for the student to work with.

The assessment server implements of two partial processes one for the Registration for the assessment and one the assessment itself. In the first, the student has to register with a certificate and therefore shows that he or she has access to the corresponding private key. The latter then secures the assessment itself. Both partial processes and other functionalities of the assessment server use also a micro service pattern to offer small, independent functionalities for different steps in the process. During the assessment, the server uses certificates from the DFN-PKI to authenticate the students. This complies on the one hand with the intended use of these certificates as documented in the certificate policy. Furthermore, the certificate policy assures that certificate authorities check ID documents when issuing a certificate (DFN-PKI, 2016).

Before the assessment, the teacher defines the students that should take part. As shown in Figure 2, the assessment server then invites unknown students to submit their certificate. If necessary, this step repeats in regular intervals until all students have registered with a certificate that expires after the date of the assessment. Students may register either by responding via a signed email or by following a link in their web browser. In the first case, the server parses the signature of the email in PKCS#7 format and extracts the contained certificate. Whereas in the latter case the web browser authenticates the user via a TLS client certificate in X.509 format (Kaur & Kaur, 2012). In either case, the server validates the certificates against the DFN-PKI infrastructure and then stores valid certificates in a key database. This database allows at later steps in the process to get easy access to students' public keys. To increase the reusability of stored certificates the certificate store extracts some metadata like validity dates, fingerprint and the public key in PEM format and saves them along with the certificate.

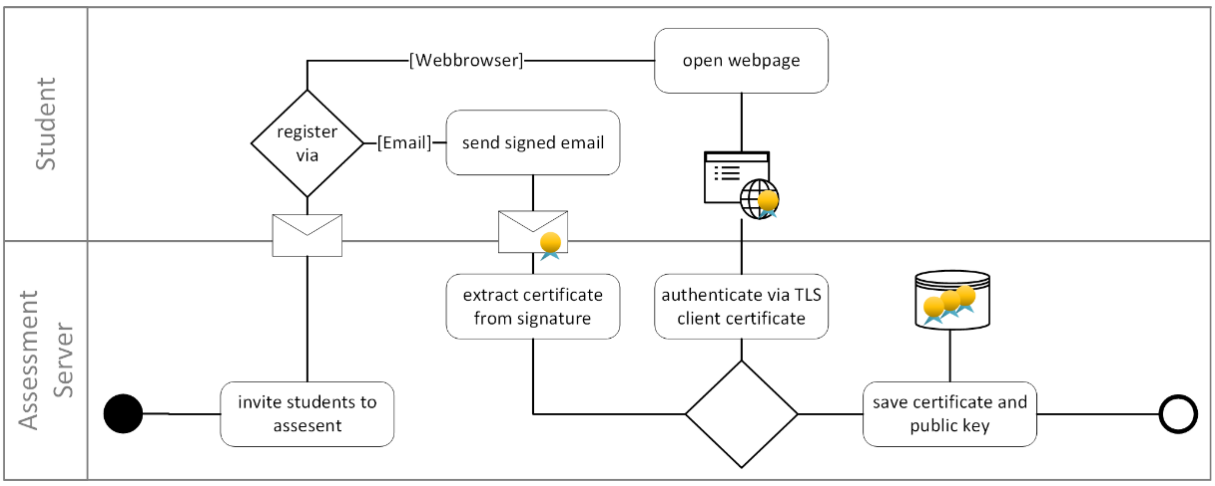

Figure 4: Partial process for assessment registration.

During the assessment, the student again authenticates with the client certificate via TLS. This time however the student uses the assessment client instead of a web browser. As Figure 5 shows, the client then tries to download the configuration for the current assessment. Upon receiving the configuration request, the assessment server validates the certificate against the certificates in its database. For each student the server creates a storage space on a compatible storage backend. Credentials needed to access the storage are stored in the configuration and then sent to the client. Furthermore, the configuration includes the task description for the assessment. With this information provided, the student can now use the client to conduct the assessment.

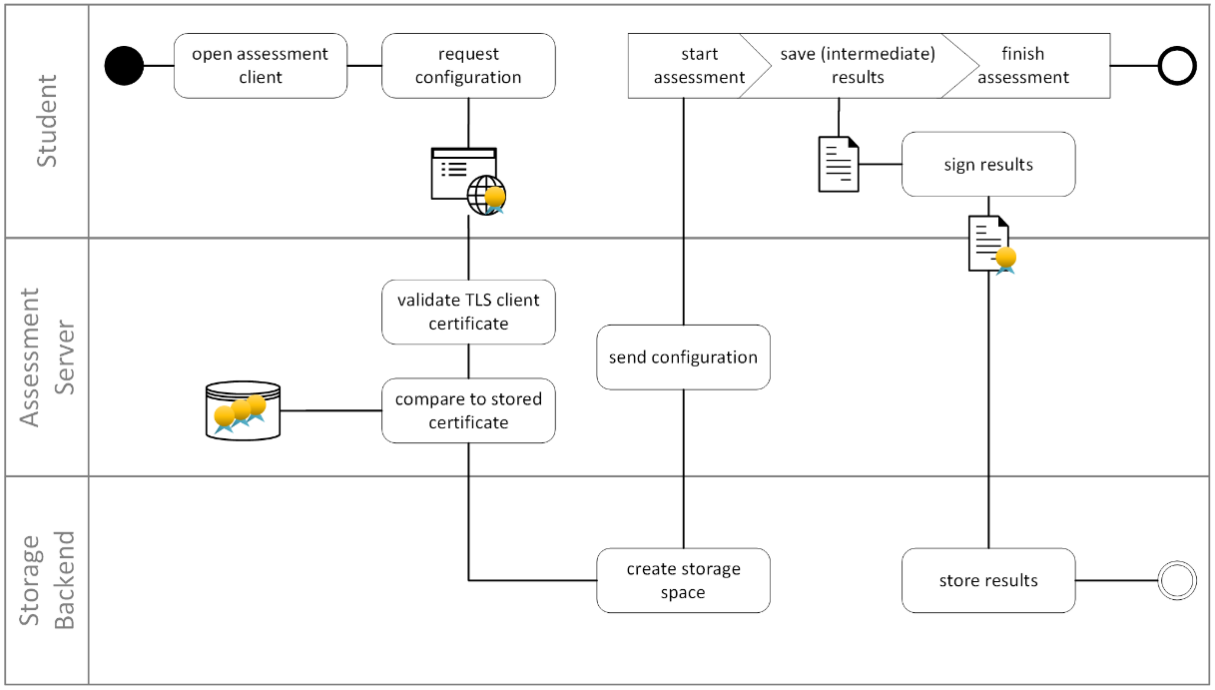

Figure 5: Partial process for taking the exam.

6. LESSONS LEARNED

During the implementation of the prototype, we noticed that the GitHub API does not have support for the GIT-native mechanism of singing GIT commits. Therefore, we decided to append the signature of the commit to the particular commit message.

Authentication using TLS client certificates only works if the server also utilizes HTTPS. While in a production environment, there is no excuse for not using HTTPS, existing development environments mostly did not offer TLS out of the box and needed configuration updates with certificates even for local testing. Opposed to the other web technologies used (IMAP, HTTP, REST), the implementation of certificate parser was complicated, libraries, like node-forge (Digital Bazaar, Inc., 2017), often do not cover full stack of parsing and interpreting different formats, certificate validation and newest algorithms.

Micro service architectures allow flexible interchange of components and easy extensibility of the system. Furthermore, they allow integration of developed services into central infrastructures more easily.

7. FUTURE CHALLENGES

Quite recently, the SHA1 hash algorithm, which is internally used by GIT, has been broken. Stevens et al. state the following regarding the affection of GIT:

"GIT strongly relies on SHA-1 for the identification and integrity checking of all file objects and commits. It is essentially possible to create two GIT repositories with the same head commit hash and different contents, say a benign source code and a backdoored one. An attacker could potentially selectively serve either repository to targeted users. This will require attackers to compute their own collision." (Stevens, Bursztein, Karpman, Albertini, & Markov, 2017)

According to that statement, the vulnerability of SHA-1 would enable an examiner to alter the contents of a student's repository after the examination. However, since the students sign their commits with their private key, the tampered repository would not contain correctly signed commits. Therefore, the vulnerability for our scenario is negligible. Additionally, since GIT is a well maintained open source project, this issue is akin to be fixed in the near future.

The DFN-PKI maintains already a public certificate directory. This could be used later on. At the moment, however, this directory has two pitfalls. First, it does not validate that a student still has access to the private key. Since the students have to use their private key in order to work with our approach, we can ensure that the student has access to it. Second, publishing one's certificate into the directory is optional, therefore students may not be in the directory, even if they have a certificate. If these issue could be resolved, due to the micro service architecture, the module for the certificate database could simply be replaced without affecting the rest of the framework.

As already mentioned in the previous section, certificate libraries were quite limited. Currently, the library used allows to parse certificates in various formats, however, it only supports RSA and not ECC. Since ECC is not as common as RSA, we still decided to use that library in order to handle the different types of certificates without problems. Finding a library that also supports ECC would be desirable.

In order to make BYOD reliable for e-Assessment, the measures presented in this paper are important parts. However, more issues have to be considered. For example, a lockdown mechanism has to be implemented, which prevents students during the assessment from unauthorized actions. These actions include, for example, starting additional software or accessing online resources.

Quite recently, the German Stifterverband granted funding to support further research of BYOD and e-Assessment (Stifterverband, 2016). Therefore, we hope that we will indeed be able to build a community, that helps to extend and improve the e-Assessment framework.

8. REFERENCES

Adams Becker, S., Cummins, M., Davis, A., Freeman, A., Hall Giesinger, C., & Ananthanarayanan, V. (2017). *NMC Horizon Report: 2017 Higher Education Edition*. Austin, Texas: The New Media Consortium.

Barata, R., Silva, S., Martinho, D., Cruz, L., & Guerra e Silva, L. (2014). Open APIs in Information Systems for Higher Education. Umea.

Biella, D., Engert, S., & Huth, D. (2009). Design and Delivery of an E-assessment Solution at the University of Duisburg-Essen. *Proceedings EUNIS*.

Birch, D., & Burnett, B. (2009). Bringing academics on board: Ecouraging institution-wide diffusion of e-learning environments. *Australasian Journal of Educational Technology*.

Boulanger, A. (2005). Open-source versus proprietary software: Is one more reliable and secure than the other? *IBM Systems Journal*, pp. 239-248.

Brauckmann, J., & Gröper, R. (2013). Konzept und Nutzen von Certificate Policy und Certification Practice Statement. *Datenschutz und Datensicherheit - DuD, 37*(8), pp. 491–496.

Bücking, J. (2010). *eKlausuren im Testcenter der Universität Bremen. Ein Praxisbericht*. Retrieved März 01, 2017, from E-Teaching.org: https://www.e-teaching.org/praxis/referenzbeispiele/eassessment-bremen

Dahinden, M. (2014). *Designprinzipien und Evaluation eines reliablen CBA-Systems zur Erhebung valider Leistungsdaten* . Zürich: ETH Zürich.

Dahlstrom, E., Brooks, C., Grajek, S., & Reeves, J. (2015). *Undergraduate Students and IT*.

DFN-PKI. (2016). *Zertifizierungsrichtlinie der DFN-PKI*.

Digital Bazaar, Inc. (2017). *npm*. Retrieved from node-forge: https://www.npmjs.com/package/node-forge

Fielding, R. T., & Taylor, R. N. (2002, May 2). Principled design of the modern Web architecture. *ACM Transactions of Internet Technology*, pp. 115-150.

Fluck, A., Pullen, D., & Harper, C. (2009). Case study of a computer based examination system. *Australasian Journal of Educational Technology*.

GitHub. (2017). *Build cross platform desktop apps*. Retrieved from Electron: https://electron.atom.io

GitHub. (2017). *The world's leading software development platform*. Retrieved from GitHub: https://github.com

GitLab Inc. (2017). *Code, test & deploy with GitLab. Everyone can contribute!* Retrieved from GitLab: https://about.gitlab.com/

Hochschulforum Digitalisierung. (2015). *E-Assessment als Herausforderung*. Essen: Edition Stifterverband - Verwaltungsgesellschaft für Wissenschaftspflege mbH.

Hochschulforum Digitalisierung. (2016). *The Digital Turn - Hochschulbildung im digitalen Zeitalter. Arbeitspapier Nr. 27*. Berlin: Hochschulforum Digitalisierung.

Janßen, D., Karami, M., Borowski, E., Richert, A., Jeschke, S., & Baumann, M. (2014). *e-Prüfungen mit dem Online-Prüfungssystem (OPS) an der RWTH Aachen University*. Berlin: GML² 2014.

Jara, N., & Molina Madrid, M. (2015). Bewertungsschema für eine abgestufte Bewertung von Programmieraufgaben. *GI-Edition Lecture Notes in Informatics Proceedings* (pp. 233-240). Bonn: DeLFI 2015 - die 13. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.

Kaur, R., & Kaur, A. (2012, September 14). Digital Signature. *2012 International Conference on Computer Sciences*, pp. 295-301.

Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview. *Theory Into Practice*, pp. 212-218.

Küppers, B., & Schroeder, U. (2016). BRING YOUR OWN DEVICE FOR E-ASSESSMENT - A REVIEW. *EDULEARN proceedings* (pp. 8770-8776). Barcelona: IATED.

Luecht, R. M., & Sireci, S. G. (2011). *A Review of Models for Computer-Based Testing*.

Michel, L. P., Goertz, L., Radomski, S., Fritsch, T., & Baschour, L. (2015). *Digitales Prüfen und Bewerten im Hochschulbereich. Arbeitspapier Nr. 1.* Berlin: Hochschulforum Digitalisierung.

Miller, B. P., Koski, D., Lee, C. P., Maganty, V., Murthy, R., Natarajan, A., & Steidl, J. (1995). *Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services.* Wisconsin-Madison: University of Wisconsin-Madison Department of Computer Sciences.

Mincer-Daszkiewicz, J. (2014). We Publish, You Subscribe — Hubbub as a Natural Habitat for Students and Academic Teachers. Umea.

Namiot, D., & Sneps-Sneppe, M. (2014). On Micro-services Architecture.

Node.js Foundation. (2017). *Node.js*. Retrieved from Node.js: https://nodejs.org/en/

Politze, M., Schaffert, S., & Decker, B. (2016). A secure infrastructure for mobile blended learning applications. *European Journal of Higher Education IT 2016-1*.

Poll, H. (2015). *Student Mobile Device Survey 2015: National Report: College Students*.

Schleicher, J. M., Vogler, M., Inzinger, C., & Dustdar, S. (2015). Smart Fabric - An Infrastructure-Agnostic Artifact Topology Deployment Framework.

Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The first collision for full SHA-1. http://shattered.io/static/shattered.pdf.

Stifterverband. (2016). *Stifterverband*. Retrieved from Fellowships für Innovationen in der Hochschullehre: https://www.stifterverband.org/digital-lehrfellows

Terzis, V., & Economides, A. A. (2011). The acceptance and use of computer based assessment. *Computers & Education*, pp. 1032-1044.

Vogt, M., & Schneider, S. (2009). *E-Klausuren an Hochschulen: Didaktik - Technik - Systeme - Recht - Praxis*.

Walker, R., & Handley, Z. (2016). Designing for learner engagement with computer-based testing. *Research in Learning Technology*.

Willige, J. (2016). *Auslandsmobilität und digitale Medien: Arbeitspapier Nr. 23*.

Winkley, J. (2010). E-assessment and Innovation. *Emerging Technologies. Conventry: Becta*.

9. AUTHORS' BIOGRAPHIES

Bastian Küppers, M.Sc. is research associate at the IT Center RWTH Aachen University. His research is focused on e-Learning and e-Assessment technologies. He received his M.Sc. cum laude in Artificial Intelligence from Maastricht University in 2012. In 2010, he finished his B.Sc. studies in Scientific Programming at FH Aachen University of Applied Sciences. Since 2010 he works at IT Center as a software developer and later as a teacher for parallel programming, robotics and other topics in computer science.

Marius Politze, M.Sc. is research associate at the IT Center RWTH Aachen University since 2012. His research is focused on service oriented architectures supporting university processes. He received his M.Sc. cum laude in Artificial Intelligence from Maastricht University in 2012. In 2011, he finished his B.Sc. studies in Scientific Programming at FH Aachen University of Applied Sciences. From 2008 until 2011, he worked at IT Center as a software developer and later as a teacher for scripting and programming languages.

Prof. Dr-Ing. Ulrik Schroeder received his Diploma degree as well as his PhD in Computer Science from Technische Universität (TU) Darmstadt. Since 2002 he heads the Learning Technologies Research Group in the computer science department at RWTH Aachen University. His research interests include assessment and intelligent feedback with a focus on learning processes, Web 2.0 applications and social software in education, mobile Internet and learning, gender mainstreaming in education, and Computer Science didactics.