Students Implement the European Student Card

by Walter Tichy

Editor's Introduction

Student mobility is a shared goal of the member states of the European Union. One ingredient that makes student mobility possible is a universal student ID card that is accepted everywhere and can be coded with services at the institutions visited. The European Student Card (ESC) is such a universal ID card, standardized in Europe. A team of students from the Karlsruhe Institute of Technology developed the software for it. It works as follows: A student with an ESC simply walks up to a self-service kiosk, presents the card to a reader, and then selects the desired services, such as cafeteria, library, lab access, etc. In this interview, the development team will explain how they made this work smoothly, including the security considerations. This project is another example of how undergraduate students can build impressive software if given a challenge, the right tools, and some supervision.

Students Implement the European Student Card

by Walter Tichy

The romantic picture of students migrating from campus to campus to meet fellow students, take courses, and perhaps have life-altering experiences is just that— romantic. In reality, undertaking such an adventure requires recognizing and coping with numerous practical details.

One major problem is registering with the different services on and around a new campus, e.g. setting up a library account or a virtual wallet to pay for meals in the student cafeteria. For this, a student would need to procure a student ID card from the university, which wouldn't happen overnight. For an informal visitor (not taking classes), this couldn't happen at all because the young scholar wouldn't even be registered.

Erasmus+ is an educational program established in 1987 by the European Union to make it easy for students to study at universities across Europe and beyond. Among other things, Erasmus guarantees students will not have to pay additional tuition fees where they visit; it also covers supplemental living expenses. Students typically stay on campus for two to three months.

Some 4,000 institutions of higher learning in 37 countries currently participate. To make exchanges as hassle-free as possible, in 2018 the European Student Card (ESC) was introduced. It eliminates the need to complete onsite registration procedures and paperwork. It allows online course registration and the automatic recognition of credits in the European Credit Transfer System. To find out exactly what the card does and the software behind it, we will talk to a team of students who implemented the ESC in a practical training course.

Team members are (from left to right in the group picture): Tilo Spannagel, Tobias Erthal, Roman Freiberg, Leon Jungemeyer, Joschka Haas, and team supervisor Marc Kiefer.

Walter Tichy (WT): Have any of you been on a student exchange, or are you planning one?

Team: Erasmus and other exchange programs have encouraged us and many of our friends to plan exchanges in the coming semesters. English speaking countries are popular choices, but many students are also considering universities in Asia.

WT: Above we cited some of the problems (other than the language barrier) you would face. How would ESC help you overcome them?

Team: The European Student Card initiative standardizes student identification cards based on RFID technology among participating universities. This means those universities will be able to safely share information about students and endow their ID cards with new services and access permissions without the need to issue new cards. In the above scenario, one would simply link

one's existing student ID card to a library account or the cafeteria wallet with the push of a button. Figure 1 shows the screen you would get by presenting your card at a kiosk.

Figure 1. A screenshot for loading services onto a European Student Card.

WT: Sounds easy, but I bet there is a lot of software and hardware hidden underneath. First, what hardware would you need to take your current ID card and code it with permissions at a foreign university?

Team: The system is designed to run on anything that hosts a Java virtual machine. Currently, our prototype uses a Raspberry Pi connected to a seven-inch touch screen and an RFID card reader. This setup could be built into a kiosk at any participating university. Here's an image of our prototype (see Figure 2).

Figure 2. Screen, Raspberry Pi, and card reader of the prototype.

WT: When you present an ID card, how does the system know it is valid?

Team: The card gets read and its unique identifier is sent to the backend service of the university running the station. This service will then contact the server that is responsible for issuing European Student Cards. This server is hosted in Paris and checks if a card is valid.

WT: Isn't there a major security risk? I could basically walk into any university and use their services, without paying tuition or being responsible for any costs or damages.

Team: Not quite. Fortunately, the European Student Card standard defines a unique identifier for each card that enables us to identify the user and therefore decide which services can be accessed and which not. The university you're visiting could decide to allow you to use their cafeteria, but not grant access to laboratories, for example.

WT: Which security measures did you build into your system?

Team: Since the system contains keys that are used to decrypt the ID cards, our first goal was to ensure that those cannot be misused to write possibly malicious data to the card whenever the card, reader, or processor are stolen. Those keys are stored in the firmware of the reader and since the firmware can't be read, they are secure there. This means the only way to misuse the keys is to use the original firmware by stealing the reader.

To prevent this, every write operation is verified by the backend service, which is run by the university. This is accomplished by using a shared secret between the card reader and the backend to compute a token that is sent to the reader for verification. When the reader is marked as stolen, such a token will not be generated and therefore the write operation will fail. The token consists of a 64-bit integer and a message authentication code of that integer generated using the shared secret. The reader will check that the message authentication code is valid, and the value of that integer must be larger than that of the last valid token. This way a token cannot be used more than once, so an attacker who captures a token cannot use it to write malicious data onto a card.

The connection between the Raspberry Pi and backend service is REST-based and uses an https connection. Thus, it is encrypted by TLS and the Raspberry Pi authenticates itself to the backend using a bearer authentication header sent with every request.

WT: What exactly is stored on a European Student Card? Just an identifier?

Team: Initially, yes. A unique ID is generated by the ESC server in Paris and stored on the card. For each new application that is enabled, the application may store an additional identifier, for

example the student account number for the cafeteria at the chosen university. There can be additional meta information, but the bulk of the data is kept on the various servers.

WT: When I visit colleagues at other universities, I sometimes would like to go to the library or a cafeteria. Could the features of the ESC be available to faculty as well?

Team: Yes, definitely, if they are issued appropriate cards, called EFCs perhaps.

WT: Great idea! Please sketch the architecture of your system.

Team: The system consists of three components: The backend system, which is responsible for validating student IDs and enumerating available services for them; the kiosk the user is interacting with to select a service; and the RFID card reader itself, which is responsible for reading and writing the card. Those components are communicating with each other using well-defined protocols over a secure connection.

WT: What libraries did you use, in particular for handling the card, security, and communication? Any problems with them?

Team: The library for handling the cards is provided by the manufacturer of the reader. For communication with the backend we used Spring and JavaFX for the GUI. The only real problem we had was with JavaFX, which didn't handle the touch input from our monitor properly. We were able to fix this by patching JavaFX.

WT: What did you learn in this project?

Team: We gained a lot of insights into how big and complex initiatives, such as the ESC, are carried out, which parties are involved, and what different kinds of considerations have to be made (political and technical).

During the development of the software we learned much about how to plan and organize software projects, especially how you continuously need to revise your drafts and make improvements.

WT: How did you communicate and share work during development?

Team: We used Slack for text-based communication and met about twice a week in person. When meeting in person was not possible, we relied on the Discord app for voice chat meetings. For sharing and collaborating during development we used Git and a university-hosted Gitlab instance.

WT: How large is the software?

Team: The software for the embedded kiosk consists of about 8,500 lines of Java and 3,000 lines of comments spread across more than 180 classes. The reader's firmware consists of about 800 lines of C.

WT: Mr. Maurer, you work at KIT (Karlsruhe Institute of Technology) in the computing center; you suggested the problem and supported the students. I can imagine universities might not be happy about "tourists" from other universities. Worse yet, aren't the team members doing something illegal? Are they heroes or hackers?

Axel Maurer (AM): Supporting mobility within the European Union is a shared, political goal of the member states, and mobility of students is part of that. The students implemented the ESC system as part of an initiative that is envisioned and supported at the highest levels of the European Commission and its Connecting Europe Facility Program (CEF). So the team members are definitely the heroes, if we want to use these categories.

Of course, there will be hacker attacks. Universities have always been exposed to the risks of students trying out what they have learned, not always with fully positive intentions. But in the end, that is a good thing and universities know how to deal with it.

WT: Are you actually planning to use the student software?

AM: Yes, definitely. There is still no final definition of the RFID-app for the ESC. As soon as it is available, we can use the software on our existing European Student Cards at KIT.

Obviously, not all card-based service systems can be adopted to accept the ESC app in the short term, so we'll always have to recode cards for some services. For example, the universities in the state of Baden-Württemberg participate in the card federation “bwCard,” which enables mutual use of the cards. For this purpose, a separate app is applied to the card, so that every student with an ESC can participate directly in all bwCard-services in the network. In the future, students can use their ESC for local university services, like the libraries or cafeterias, in the state of Baden-Württemberg without administrative overhead.

We will set up the system in the library network of KIT. Other universities including the Universities of Konstanz, Freiburg, and Stuttgart have also asked for the system.

WT: So the students actually obtained a commitment for adoption. This is wonderful, because without adoption, innovation will not happen. Final question: Will Brexit affect the ESC?

AM: No, the ESC is valid in the European Higher Education Area (EHEA). EHEA currently covers 48 countries and the U.K. has no intention to leave it.

Acknowledgements

Thanks to Phil Yaffe for contributing a smooth introduction.

About the Author

Walter Tichy has been professor of Computer Science at Karlsruhe Institute of Technology (formerly University Karlsruhe), Germany, since 1986. His major interests are software engineering and parallel computing. You can read more about him at www.ipd.uka.de/Tichy.

DOI: 10.1145/3388701