Zeitschrift: Visionen : Magazin des Vereins der Informatik Studierenden an der ETH Zürich

Herausgeber: Verein der Informatik Studierenden an der ETH Zürich

Band: - (2017)

Heft: 4

Heft

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 20.08.2025

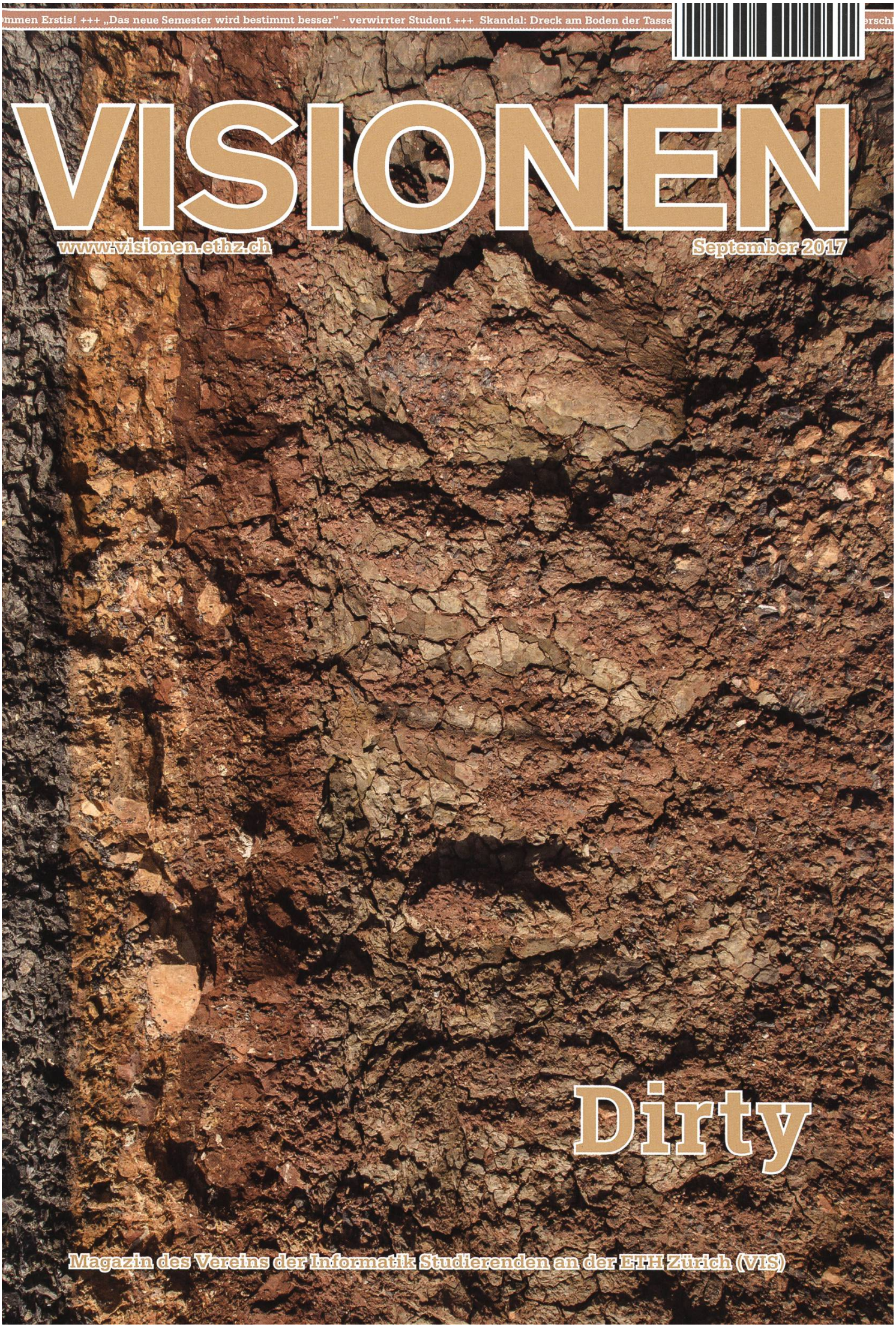
ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

mmen Erstis! +++ „Das neue Semester wird bestimmt besser" - verwirrter Student +++ Skandal: Dreck am Boden der Tasse erschi

VISIONEN

www.visionen.ethz.ch

September 2017



Dirty

Magazin des Vereins der Informatik Studierenden an der ETH Zürich (VIS)

open
systems

Open Systems gehört mit seinen Mission Control Security Services im Bereich IT-Sicherheit zu den europaweit anerkannten Anbietern. Wir arbeiten von Zürich und Sydney aus in einem dynamischen Umfeld in über 180 Ländern. Bei uns kannst Du Dein Wissen in einem jungen Team in die Praxis umsetzen und rasch Verantwortung übernehmen. Infos über Einstiegs- und Karrieremöglichkeiten sowie Videos findest Du auf unserer Website. **www.open.ch**

MOST ATTRACTIVE EMPLOYERS
SWITZERLAND
universum
2017

Editorial

Liebe Leserinnen und Leser

Ich hoffe, eure Prüfungen sind gut gegangen. Während ich diese Zeilen schreibe, liegt die Prüfungssession allerdings noch in gefühlt weiter Ferne. Dass das vermeintlich Ferne in der Realität allerdings auch bedeutend näherliegen kann, erfuhr Zeno mit dem Fahrrad. Liest man seinen Reisebericht, scheinen Passfahrten von Graubünden nach Italien ein Zuckerschlecken. Umso befriedigter und erholter, stelle ich mir vor, kommt man nach einer solchen Tour aus den Ferien zurück.

Da dies die erste Ausgabe des Herbstsemesters ist, möchte ich natürlich alle Erstis herzlich willkommen heissen. In dieser Ausgabe geben wir euch viele nützliche Tipps an die Hand und besonders wichtig, einen Steckdosenplan für die Hörsäle, in denen ihr eure ETH-Karriere beginnen werdet. Ihr merkt früh genug warum.

Hier möchte ich euch auch auf den VIS aufmerksam machen, der beste Studentenverein der ETH (und der Welt?). Der VIS hilft in allen Lebenslagen, sei es mit gratis Kaffee am Morgen oder wenn eine Vorlesung mal nicht so der Brüller ist. Wie der VIS dort hilft und mit Professoren zusammenarbeitet, wird im ASL-Artikel beschrieben.

Zu guter Letzt möchten wir in dieser Ausgabe aber auch noch ein bis zwei Kontrastpunkte haben, darum: Wenn es im Aufenthaltsraum zu «dirty» wird und eine Auswahl «dirty secrets» aus der neueren Geschichte.

Ich wünsche euch einen sauberen Semesterstart und viel Spass bei der Lektüre dieser Ausgabe des Visionen.

Euer Chefredaktor

J. Croci

Julian Croci

Inhalt

Dirty

I'll tell you my dirty little secret… 6

Ordnung – aber wie? Eine Schritt-für-Schritt-Anleitung zum Aufenthaltsraum 8

Offizielles

Leserbrief: Warum verständlich wenn's auch kryptisch geht? 12

State of ASL 14

Berichte

Things I Wish I Knew When I Started Studying Computer Science 18

ASL Tips 22

Reisebericht: Zernez – Trento 26

Build Your Own Multimedia Center 32

Syncthing 38

Ein Tag im Leben eines Chemiestudenten 46

Serien

Never Heard of It #18 47

Beni Koller: Zunderspiele 50

„Become part of the Sensirion success story". Wollen Sie Ihrer Karriere den entscheidenden Kick geben und sich neuen Herausforderung stellen? Dann heissen wir Sie herzlich willkommen bei Sensirion.

Sensirion steht für Hightech, Innovation und Spitzenleistungen. Wir sind der international führende Hersteller von hochwertigen Sensor- und Softwarelösungen zur Messung und Steuerung von Feuchte, Gas- und Flüssigkeitsdurchflüssen. Unsere Sensoren werden weltweit millionenfach in der Automobilindustrie, der Medizintechnik und der Konsumgüterindustrie eingesetzt und tragen zur stetigen Verbesserung von Gesundheit, Komfort und Energieeffizienz bei. Mit unserer Sensorik liefern wir damit einen aktiven Beitrag an eine smarte und moderne Welt.

Schreiben Sie Ihre eigenen Kapitel der Sensirion Erfolgsgeschichte und übernehmen Sie Verantwortung in internationalen Projekten. Stimmen Sie sich auf www.sensirion.com/jobs auf eine vielversprechende Zukunft ein.

www.sensirion.com/jobs

SENSIRION
THE SENSOR COMPANY

I'll tell you my dirty little secret…

SARAH KAMP – WÜHLT SICH DURCH SKURRILE VERSCHWÖRUNGSTHEORIEN

Vor allem die US-Regierung landet gerne mal wegen pikanter Enthüllungen in den Schlagzeilen. Ob diese erst nach Jahrzehnten oder direkt im Anschluss enthüllt wurden – bei allen wollte es keiner glauben, bis es rauskam. Hier meine Favoriten:

Operation Northwoods

Wir schreiben das Jahr 1962, die USA betreibt verdeckte Kriegsführung gegen Kuba, John F. Kennedy ist seit einem Jahr im Amt des Präsidenten. Nachdem im April 1961 die Invasion in der Schweinebucht so kläglich gescheitert ist, hat Kennedy keine politische Entschuldigung mehr, die bereits bereitstehende US-Marine in Kuba eingreifen zu lassen. Daher sucht er nach einer anderen, subtileren Möglichkeit, eingreifen zu dürfen, da auch die Beteiligung der CIA bei der Invasion bekannt geworden ist. Unter dem Decknamen «Operation Northwoods» wurde bereits unter Präsident Eisenhower ein Geheimplan verfasst, der Kennedy nun vorgelegt wird. Der Plan umfasst eine Reihe von Terroranschlägen gegen den US-eigenen Luft- und Schifffahrtsverkehr innerhalb des Landes, die man im Nachhinein dann Fidel Castro zuschieben möchte. Ausgehend von der Annahme, dass die Bevölkerung der USA einen militärischen Angriff nur als Reaktion auf bedrohliche Aktionen seitens Kuba gegen amerikanische Soldaten und Zivilisten unterstützen würde, war das Dokument von allen Mitgliedern der Vereinigten Stabschefs und dem Vorsitzenden der NATO in Europa unterzeichnet. Durchgeführt wurde die Operation nie, da Kennedy sich weigerte, den Plan auszuführen. An die Öffentlichkeit kam das ganze erst 1997 durch den Freedom of Information Act.[1]

Durchgeführt wurde die Operation nie.

MKULTRA

1953 befinden sich die USA im Kalten Krieg. Einen wichtigen Teil der Arbeit eines Geheimagenten nimmt die Gefangennahme und das Verhören von Sowjetspionen ein. Und wie es sich für ein ordentliches Verhör gehört, soll dabei auch ein Wahrheitsserum verwendet werden, an welchem ein Forschungsteam im Auftrag der CIA intensiv arbeitet. So hilfreich Laborratten auch sind, kommt man bei dem Versuch der Gedankenkontrolle durch Halluzinogene wie LSD schnell an die Grenzen seiner Möglichkeiten. Daher macht man sich auf die Suche nach Versuchspersonen, die sich vorzugsweise nicht dagegen wehren können. Zwischen 1953 und den 1970er Jahren wurden tausende von Menschenversuchen an Krankenhauspatienten und Gefängnisinsassen durchgeführt. Die Testpersonen trugen bei den Experimenten schwerste körperliche und psychische Schäden davon. Ganz offensichtlich verstiess man damit gegen US-Gesetze, doch den grossen Durchbruch erzielte man damit offenbar trotzdem nicht. Die CIA bezeichnet im Nachhinein die Ergebnisse dieser Forschung als wertlos, da die Leiter der Experimente häufig ohne jede wissenschaftliche Qualifikation waren. Mitte der 70er Jahre wurden dann fast alle Akten zu diesem Projekt vernichtet, was die Untersuchung und Veröffentlichung des Falls stark erschwerte.[2][3]

Wie es sich für ein ordentliches Verhör gehört, soll dabei auch ein Wahrheitsserum verwendet werden.

Operation Mockingbird

Zu der beeindruckenden Unternehmensgeschichte der CIA gehört auch die «Operation Mockingbird». Um die öffentliche Meinung zu beeinflussen, fing man in den 1950er Jahren an, ein Netzwerk aus führenden amerikanischen Journalisten aufzubauen, um die Sichtweise der CIA möglichst gezielt zu verbreiten. Zu dieser Propaganda-Aktion gehörte auch die Gründung von Kulturvereinen oder Studentenvereinigungen. Nachdem das Projekt mit der Zeit immer grösser wurde, griff die CIA auch auf ausländische Journalisten und politische Kampagnen über. Diese Manipulation der Öffentlichkeit wurde von der CIA inzwischen offiziell anerkannt und die verbleibenden Akten dazu wurden freigegeben, jedoch sei bloss ein siebenseitiges Dokument dazu übrig geblieben.[4]

Quellen

- http://nsarchive.gwu.edu/news/20010430/doc1.pdf
- http://documents.theblackvault.com/documents/mkultra/mkultraindex.pdf
- http://www.theblackvault.com/documentarchive/cia-mkultra-collection/
- http://documents.theblackvault.com/documents/cia/operationmockingbirdCIA.pdf

Ordnung – aber wie?
Eine Schritt-für-Schritt-Anleitung zum Aufenthaltsraum

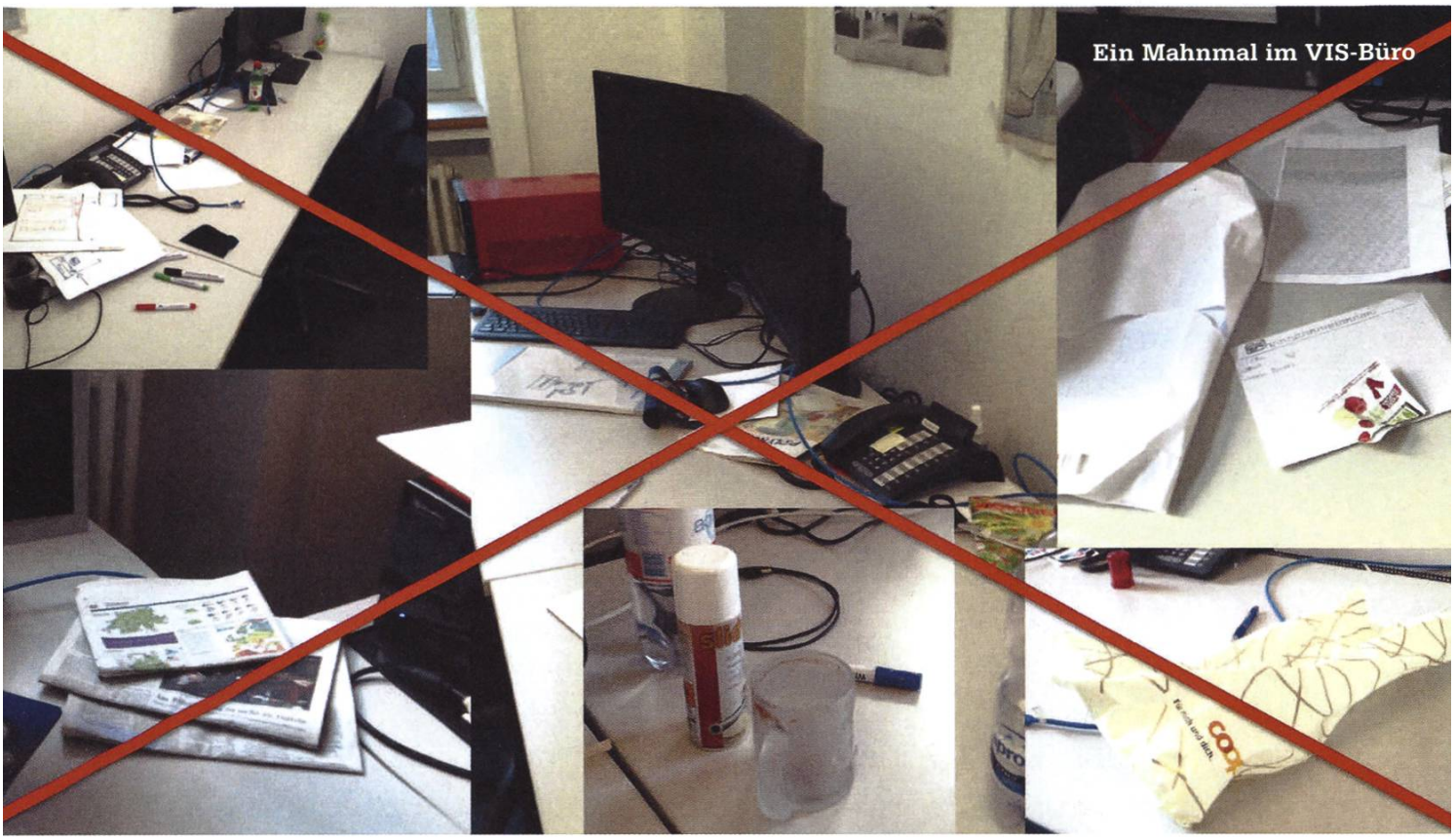
ANDREAS BROMBACH – MAG ORDNUNG, RÄUMT ABER NICHT GERNE AUF

Hallo, Ersti. Zunächst einmal herzlich willkommen an der ETH, viel Spass und Erfolg im Studium, bla bla bla und so weiter. Da du höchstwahrscheinlich schon genug halbherzige Begrüssungsfloskeln gehört und gelesen hast, möchte ich diesen Teil gerne hinter mir lassen und auf den wichtigsten Bestandteil des Studiums zu sprechen kommen: den studentischen Aufenthaltsraum. Mir ist egal, wie deine guten Vorsätze lauten; immer schön deine Serien lösen, jede Vorlesung besuchen. Vielleicht bist du ja wirklich motiviert genug, um dies auch bis zum Semesterende durchzuziehen. Doch seien wir mal ehrlich: spätestens durch dieses Magazin hast du mindestens einmal vom VIS[1] und all seinen Angeboten gehört. Da mag irgendwann in der Hälfte des Semesters ein Tischfussballmatch oder eine Runde Billard durchaus eine verlockende Alternative zur öden Vorlesung sein. Falls du noch immer tapfer gegen rapide sinkende Motivationswerte anzukämpfen versuchst, kommt vielleicht eine Tasse Tee oder Kaffee in der Mittagspause gerade recht. Zumindest verkündete einst ein grosser Herrscher: (nicht ganz exakt überliefert) «Viele Wege führen zum Aufenthaltsraum».

Hallo, Student aus dem höheren Semester.[2] Schön, dass du immer noch die Zeit findest, um dieses Heft zu lesen. Mit Sicherheit hast du dich hier schon längst gut eingelebt, verbringst inzwischen auch weite Teile deiner immer knapper werdenden Freizeit an der ETH und Umgebung und hast natürlich deine personalisierte VIS-Tasse im Regal stehen. Was ist jetzt, abgesehen von tonnenweiser Schleichwerbung, der Sinn dieses Artikels? Offensichtlich haben wir doch, im Gegensatz zu anderen Fachvereinszeitschriften, bereits genug andere Artikel in dieser Ausgabe, sodass solch ein unter Schlafdeprivation verfasster Text nun wirklich nicht mehr nötig wäre. Nun, die Angebote des VIS sind allesamt ehrenamtlich organisiert und somit entweder gratis oder nicht gewinnorientiert. Das bedeutet meiner Meinung nach nicht, dass die Dinge zur Selbstverständlichkeit werden sollten, nur weil man keine Gegenleistung erbringen muss. Diese Haltung entwickelt sich schnell

Was ist der Sinn dieses Artikels?

HALT' UNSER oVIS SAUBER!


Ein Mahnmal im VIS-Büro

zur Faulheit. Und genau diese Faulheit ist es, die den Aufenthaltsraum schon mehr als einmal in einen einzigen Haufen Chaos verwandelt hat. Exemplarisch hierfür seien genannt: im Waschbecken oder auf der Kaffeemaschine abgestellte Tassen, dreckiges Geschirr im Waschbecken, Flaschen und sonstiger Abfall auf den Tischen oder Müll, der von sich selbst überschätzenden Hobbybasketballern in Richtung Eimer geschleudert wurde und nun als Mahnmal der nicht vorhandenen Treffsicherheit hinter oder neben den Tonnen landet.

Egal ob du nun am Anfang deines Studiums stehst, voll motiviert und aufnahmefähig für Neues bist, oder schon öfter von semi-seriösen Ratgebern belästigt wurdest, sind hier ein paar wirklich einfache und für jeden Wissensstand geeignete Tipps, wie du dir, deinen Mitmenschen, dem Putzpersonal und dem

Fussnoten

[1] Wirklich nicht? Also gut. VIS steht für Verein der Informatik Studierenden, der offiziell als bester Fachverein der ETH gilt. Ist hauptsächlich als Kaffeelieferant bekannt, weiterhin Organisator für tolle Events, Ansprechpartner in fast allen Lebenslagen, psychologische und physiologische Beratungsstelle und Verkaufsstelle für grossartiges Merchandise.

[2] An dieser Stelle müsste eigentlich «Studierender/in», «Student*in», «Studierx» oder eine andere zeitgemässe Sprachverhunzung stehen. Doch meine tief verwurzelte Abneigung gegen dieses Luxusproblem würde eine weitere Flut wirrer Gedankengänge nach sich ziehen, die mit dem eigentlichen Artikel nichts mehr zu tun haben. Wer sich an dieser Stelle angegriffen fühlt, darf dies gerne persönlich mit mir austragen.

→

So nicht!

Infrastrukturvorstand das Leben im Aufenthaltsraum um einiges angenehmer gestalten kannst. Nachdem du deinen Kaffee, Tee oder dein Getränk deiner Wahl getrunken hast, wasche deine Tasse ab (optional) und stelle sie wieder zurück ins Regal (obligatorisch). Dies erspart dir nicht nur die langwierige Suche zu Beginn der nächsten Kaffeepause, denn in fast allen Fällen befindet sich das Trinkgefäss am selben Ort, an dem du es zuletzt abgestellt hast. Es hilft auch, um sich die richtigen Begriffe der verschiedenen Möbel im Gedächtnis zu halten. Ein Waschbecken ist kein Regal, folglich sollte man darin seine Tasse nur waschen und nicht abstellen. Die Rückrichtung gilt analog, auch wenn sich die praktische Umsetzung als schwieriger erweist.

Nur Kaffee und Tee zu trinken, wäre auf Dauer langweilig. Im Aufenthaltsraum sind daher Getränke aller Art ausdrücklich erlaubt. Damit dies weiterhin so bleibt, sollten aber eventuelle Abfälle an ihren richtigen Entsorgungsort gebracht werden. Auch hier gibt es mehrere idiotensichere Konzepte wie klar als solche markierte PET- und Alu-Entsorgungsbehälter, ein genau spezifiziertes System für Pfandflaschen und Altglas sowie an strategisch günstigen Orten aufgestellte Mülleimer.

Als echter Geheimtipp gilt der zweite PET-Container, der nur zwei Meter vom ersten entfernt ist. Ist nämlich der erste wieder mal komplett überfüllt, kann man entweder viel Kraft und Zeit aufwenden, die leere Flasche doch noch in den überquellenden Container zu drücken oder aber man läuft einfach zwei Meter weiter, wo ein Container wartet, der in fast allen Fällen noch genügend Kapazität bietet.

Als bester Ordnungshelfer wurden die Wandhalterungen am Waschbecken nominiert.

Als bester Ordnungshelfer wurden einstimmig die Wandhalterungen am Waschbecken nominiert. Über und neben dem Becken platziert, bieten sie ein warmes und gemütliches

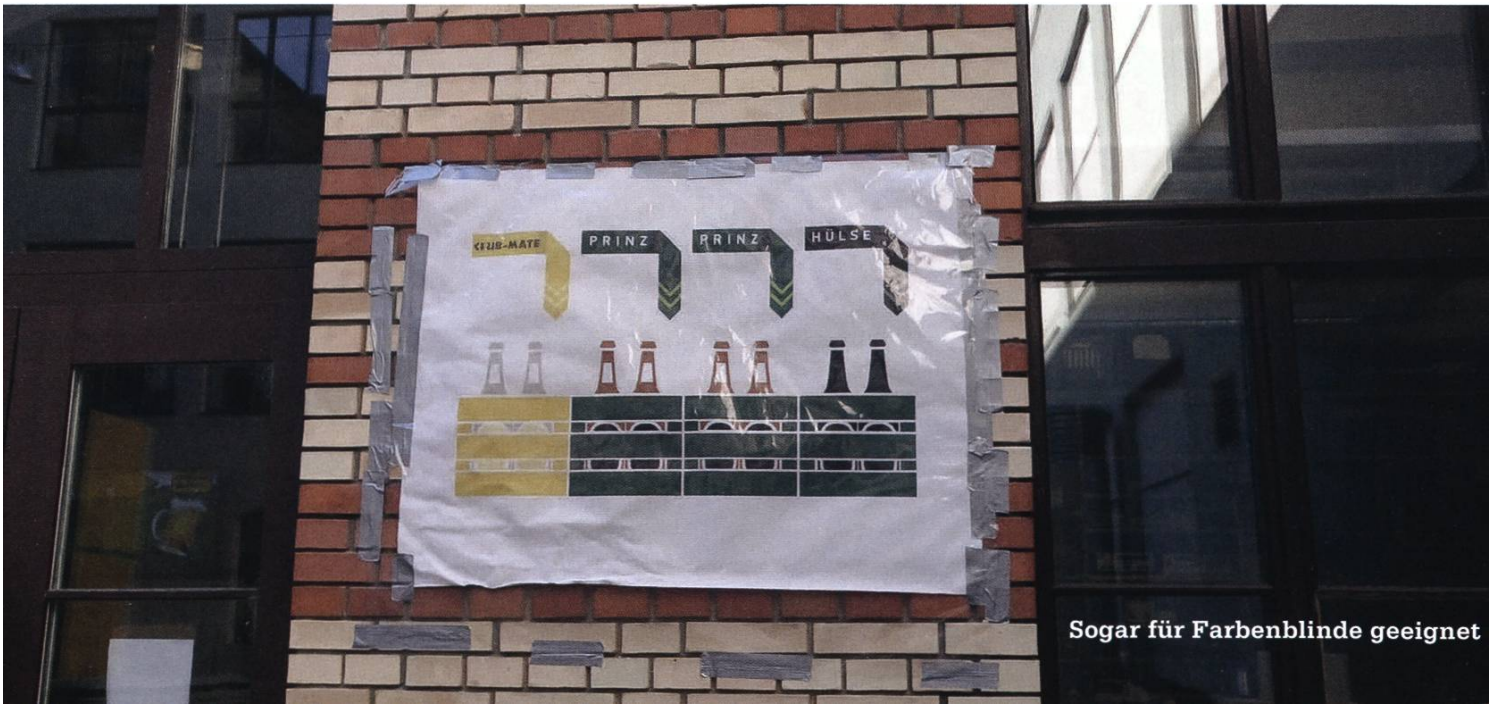
Plätzchen für all die unermüdlichen Helfer, die deine Tasse und Geschirr von Schmutz befreien. Wenn du also eine Bürste, eine Spülmittelflasche, einen Schwamm oder ein Handtuch hilflos herumliegen siehst, nimm dir doch zwei Sekunden Zeit, um sie wieder heil an ihren angestammten Platz zurückzubringen. Man wird dir dafür dankbar sein.

Im Bücherregal stehen so manch interessante Dinge herum. Von durch Studenten aus dem höheren Semester gespendeten Lehrmitteln, über Programmierbücher längst vergessener und verdrängter Sprachen, bis hin zu gewöhnlicher Unterhaltungslektüre findest du hier sicherlich etwas, das deinen Geschmack trifft. Du bist natürlich herzlich eingeladen, selbst etwas beizusteuern, um das Prinzip einer Tauschbörse aufrecht zu erhalten. Wenn alle Benutzer nur leechen, aber nicht seeden, kann das System nicht funktionieren.[1] Nach der Lektüre eines Buches empfiehlt es sich, dieses wieder an seinen Platz im Regal zurückzustellen und nicht etwa auf dem Tisch oder Sofa liegen zu lassen.

Die Liste liesse sich jetzt endlos weiterführen, doch ich hoffe, dass du das Prinzip verstanden hast. Der Rest ist trivial und wird dem Leser als Übungsaufgabe überlassen.

Fussnoten

[1] Hier nur in Bezug auf das Bücherregal. Wenn du deine Torrents an der ETH seeden möchtest, könntest du eventuell Ärger bekommen.



Sogar für Farbenblinde geeignet

Leserkommentar zum Artikel „Linus Thorvalds' Linked List" in der Juni-Ausgabe der Visionen.

Warum verständlich wenn's auch kryptisch geht?

JÜRG GUTKNECHT, 15. JUNI, 2017

Diese Variation eines bekannten Bonmots dominiert bei der Lektüre des Beitrages von Jonathan Unger[1] unweigerlich meine Sinne und erstickt jegliche Bemühungen, Wertschätzung für das „elegante" Thorvalds'sche Programm zum Entfernen eines Elementes in einer einfach verketteten Liste aufzubringen im Keime. Zwar weist der Thorvalds'sche Vorschlag gegenüber dem „naheliegenden" Code des Autors zwei offensichtliche Vorteile auf: er kommt a) mit einer einzigen Laufvariablen und b) ohne if-Anweisung aus. Doch steht demgegenüber der schwergewichtige Nachteil des Bruches der Abstraktionsebene aller modernen Programmiersprachen, welche Adressen auf Referenzen meiden wie der Teufel das Weihwasser. Dass es mit denselben Vorteilen aber ohne Abstraktionsbruch auch geht, zeigt das folgende Codefragment:

```
cur = head;
while (cur.next != entry) cur =
cur.next; cur.next = entry.next;
```

Neben der impliziten Voraussetzung, dass das zu entfernende Element entry in der Liste tatsächlich vorkommt, wird bei diesem «vorausschauenden» Algorithmus allerdings vorausgesetzt, dass der Listenkopf durch ein Dummy Element dargestellt wird, durch ein so genanntes Sentinel, dessen Nützlichkeit in jeder anständigen Anfängervorlesung doziert wird.

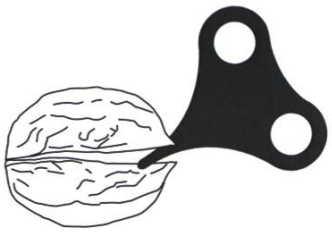
Ich kann es mir abschliessend nicht verkneifen meinem Missfallen gegenüber der Faszination der im Thorvald'schen Code zelebrierten „Cleverness" (Zitat Autor) Ausdruck zu verleihen, ist es doch gerade diese Art von kryptischem, nur Anhängern des zugehörigen Kults verständlichem Code, welche in der Masse und Kombination massgeblich zur IT Branchenmisere in Form von gescheiterten Projekten und unzuverlässigen Dienstleistungssystemen beiträgt. Ich wünschte mir dieselbe Faszination ausgestrahlt von Klarheit und wahrer Eleganz, insbesondere wenn das Motto „Um schönen Code bemüht" lautet.

Referenzen

[1] siehe Visionen 2017-03

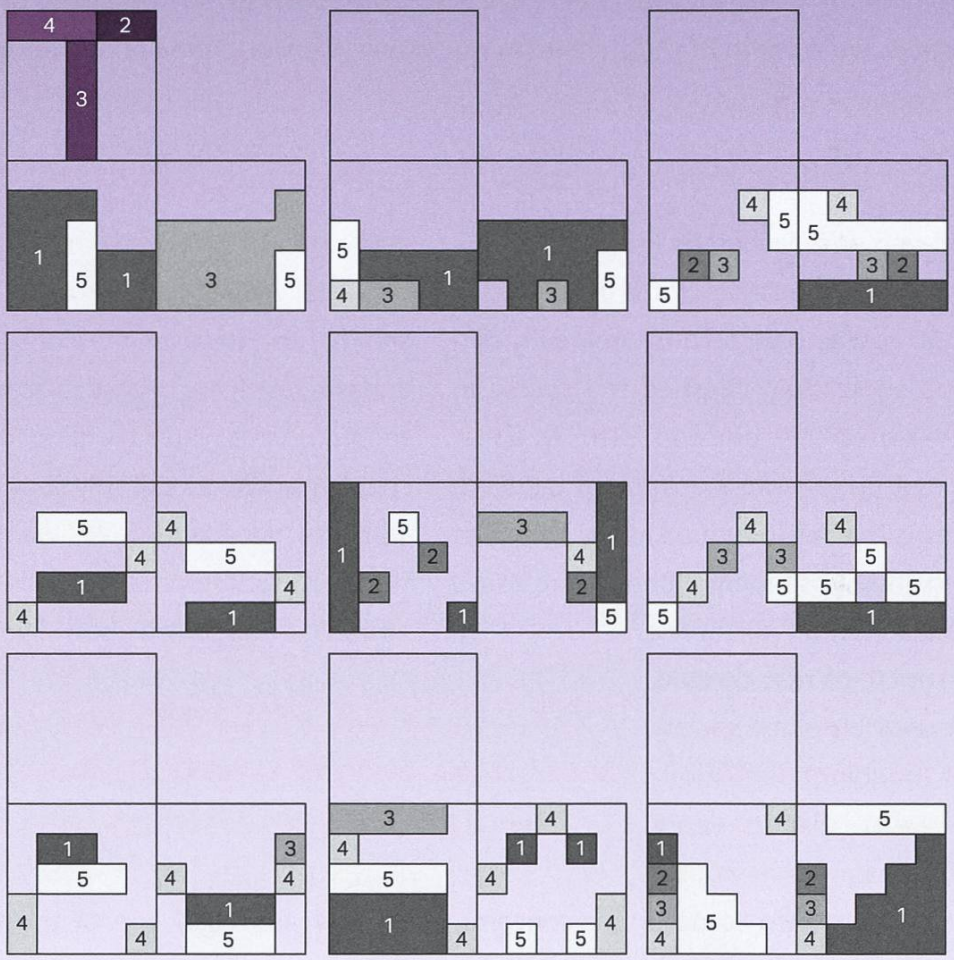
Knacknuss

präsentiert von Ergon Informatik AG



Würfel in Normalprojektion

Im Kino ist 3D schon passé. Bei den Rätseln aber extrem angesagt – oder das wird es zumindest sein nach dieser Knacknuss-Ausgabe. Die Figuren bestehen aus Würfelchen in Normalprojektion. In der Aufsicht zeigt sich jeweils ein Zeichen. Die Zahlen bezeichnen die «Höhe» (Abstand von der Ursprungsebene). Das herausgeknobelte Wort ist chiffriert. Damit du es lesen und einsenden kannst, musst du es erst decodieren. Dabei hilft dir ergon.ch/code. Am gleichen Ort erhältst du auch einen Hinweis zur korrekten Würfelbetrachtungsweise. Das endgültige Lösungswort hat unterschiedliche Bedeutungen. Mein Favorit daraus ist klar.



Rätselautor: Erwin Huber
Entwicklungsleiter
Web Application Security

Unter allen richtigen Einsendungen verlosen wir eine Hunderternote.

Lösung bis 11. November 2017 mit Name, Adresse, Name der Hochschule und Alter an visionen@ergon.ch senden.

ergon
smart people – smart software

Teilnahmeberechtigt sind Studierende mit gültigem Ausweis. Die Gewinner werden ausgelost und von Ergon Informatik AG per E-Mail benachrichtigt. Ihre Daten werden vertraulich behandelt und nicht an Dritte weitergegeben. Der Rechtsweg ist ausgeschlossen. Mitarbeitende von Ergon Informatik AG sind vom Wettbewerb ausgeschlossen.

State of ASL

HOPO ASL TASKFORCE – MEDIATES INTENSIVELY

During the past years, objections have been raised regarding the Advanced Systems Lab (ASL). Despite significant changes to both the structure and the content of the course, the feedback to the official course evaluation turned out to be disappointing. In response, Prof. Alonso and Dr. Agarwal, who is responsible for education development at the department, reached out to the HoPo commission of the VIS. In the following, we–the HoPo-internal ASL taskforce–aim to give you a short overview of our past and current activities. In the remainder of the article, we want to highlight the–in our view– positive changes to the course.

After being approached by Prof. Alonso, the HoPo team assembled a task force consisting of former ASL course participants, including students who successfully finished the course in 2015 or repeated the course in 2016. As a first step, we met up to write a document detailing the concerns that we identified as the source of frustration. In a subsequent meeting, the issues were discussed with Prof. Alonso.

We do not think that it would be fair or useful to share specifics of the (ongoing) discussion. However, we are glad to report that Prof. Alonso was very open to criticism and shared some of our concerns. Given the improvements from 2015 to 2016, we are confident that the students' concerns are being taken seriously and that, collectively, we will find a way to mitigate the existing problems.

We are confident that we will find a way to mitigate the existing problems.

If you are a former ASL course participant, we would like to take the opportunity and invite you to join our efforts. The HoPo team works hard to improve the students' conditions. For example, it is not least due to their efforts that, starting from autumn semester 2017, the master's labs will account for eight ECTS points instead of six. However, the HoPo team in general, and this task force in particular, require a diverse set of students who are willing to share their perspective and work out solutions in collaboration with the department. We want to emphasize that feedback from students is taken seriously and, in general, acted upon. If you are interested in joining the team, we would like you to drop a note to the HoPo mailing list[1]–or approach one of the team members directly.

Returning to the subject of this article, we believe that it is possible to provide a lab course in the area of systems research that is both challenging and motivating. Besides discussing existing concerns, those in our team who retook the course last year were also able to name improvements over the course from one year earlier. We are convinced that the most recent

changes to the course are a step in the right direction. In the remainder, we would like to highlight some of the changes that were made and give reasons why we think they improve the situation.

Improvements over 2015

Three instead of Two Milestones

The project is comprehensive in that it covers everything from the implementation of a middleware to the analysis and modelling thereof. The analysis entails the deployment of the system, writing and deploying a testing infrastructure–a task at least as time-consuming as the implementation. In each phase of the project, there are many degrees of freedom where you have to find the right trade-offs.

Prior to 2016, all of the above were packed into two large milestones. In complex projects, it is important to have short feedback cycles. While the decision to have three instead of two milestones also puts more pressure on the course staff, the benefit for the students cannot be overstated.

No Intermediate Deployment

In 2015, students were asked to deploy and test their system on a computer cluster here at ETH before deploying it to the actual test environment on Amazon Web Services. Getting the system to run in either of the two environments is time-consuming. As the configurations of the two environments differ considerably, the experience was difficult to translate from one to the other. In 2016, this intermediate step was left out, saving the students a lot of time–we reckon the same is true for the assistants.

Architecture Template

In the years prior to 2016, it was left to the students to come up with an architecture for the system under test. In some cases, this resulted in hopelessly overcomplicated designs that turned out to be difficult to analyze and model.

In 2016, a high-level architecture was proposed by the course staff. The architecture template acted as a common reference point when discussing one's own solution with the course staff and generally reduced ambiguity.

Predefined Workload and Components

Prior to 2016, students did not only need to write the middleware, but also the load-producing clients. The database system had to be set up and configured properly. Even though a standard database system was used, configuring it gave rise to some subtle mistakes. For example, given the nature of the problem to solve, if you forgot to adjust the isolation level properly, you would often end up having bogus results; depending on the workload, you would have to reconfigure the database's internal garbage collector, etc.

Besides massively reducing students' workload, using a predefined client and server also clarified what protocol features the middleware had to support and gave a solid basis for testing the middleware. It also clarified what type of workload was to be used.

Implementation is Valued

Even though the Advanced Systems Lab is not a software engineering course, last year, points were granted for a correct implementation in the first milestone. Given the fact that the implementation of the system takes up a significant portion of the project, we think this is justified. ➔

Conclusion

At the end of the day, neither do students want to devalue the master's degree by making courses easier, nor is it the intention of the course staff to treat students unfairly. We believe that by working together, resolving ambiguities and misunderstandings (with respect to both the course content and the discussion surrounding it), a systems lab course will emerge that is challenging, motivating, and fair. The improvements we have seen in the year 2016 further strengthen our conviction in this regard and we are looking forward to continuing the discussion. After all, there is no reason why the ASL could not be the most popular lab in the master.

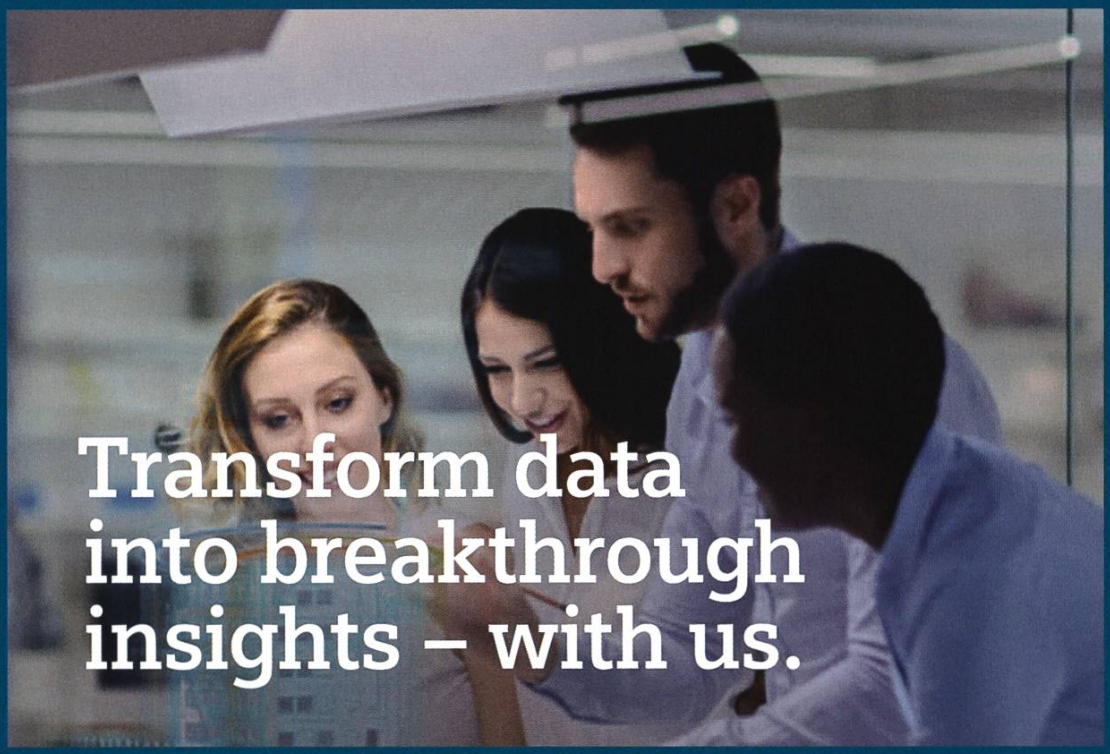
Footnotes

[1] hopo@vis.ethz.ch

Bildnachweise

Cover: © Fotolia
S. 8-11: © VIS
S. 26-30: © Zeno Koller
S. 38-45: syncthing.net
S. 47: sidewalksandskeletons.bandcamp.com/
S. 48: © VIS

SIEMENS
Ingenuity for life



Looking for new challenges every day? Want to work alongside with the sharpest minds in your field? Welcome at Siemens.

We are searching ambitious people all across the world:
- making things talk with IoT: develop, deploy and run digital services, create your own applications, or even new business models
- making buildings and cities smarter
- enhancing travel efficiency and comfort through digitalization
- breaking world records with software and system engineering

Visit
siemens.ch/jobs

Things I Wish I Knew When I Started Studying Computer Science

ALINE ABLER – VERANSTALTET GERNE GOOGLE-DOC-PARTYS

TL;DR der VIS ist awesome!

- Es gibt einen Gang, der die Mensa und das HG verbindet.
- Projekt Neptun verkauft vergünstigte Laptops an Studierende - das Verkaufsfenster ist noch bis zum 2. Oktober offen.
- Es ist okay, wenn du es nicht auf den ersten Versuch schaffst.
- In der VIS Lounge im CAB gibt es für Mitglieder gratis Kaffee! Wer keinen Kaffee mag, kann auch Tee oder ein Bier haben.
- Im CAB gibt's das Relax, ein Ruheraum für Powernaps.
- Theorie lernen ist gut, Übungen lösen ist besser.
- Auf der Dachterrasse des CHN hat es einen Grill.

Auch höhere Semester verlaufen sich noch ständig im HG.

- Git ist sehr nützlich, wird einem aber nie wirklich beigebracht.
- Später im Studium wirst du Linux brauchen - wenn du also Zeit hast, probier es am besten jetzt schon aus, damit du dann schon klar kommst.
- Es gibt eine studentische Organisation an der ETH für Linux-Support (TheAlternative.ch), die auch jedes Semester Linux-Kurse anbietet.
- Im "Hot Pasta" gegenüber vom CAB kriegt man einen Franken Rabatt, wenn man ein eigenes Tupperware mitbringt.
- Im Uni-Lichthof gibt es gute Snacks und auch Mittagessen, meistens Pommes.
- Der VIS beisst nicht. Die Leute da sind meistens ganz lieb und beantworten Fragen.
- Der AMIV verkauft im Sommer Glace. Einfach in deren Büro im CAB vorbeigehen und fragen.
- Grundrisse von allen ETH Gebäuden kann man hier finden: https://www.ethz.ch/services/de/service/raeume-standorte-transporte/raeume-gebaeude/gebaeudeorientierung.html
- Jimmy's Pizza ist recht nah und hat abends sehr gute Pizzen zum Studentenpreis für 15.- .

- Vernachlässige deine persönlichen Code-Projekte nicht - sonst macht Informatik keinen Spass mehr.
- Durch die Mathe musst du durch - die spannendsten Fächer kommen erst später.
- Wer schon gut programmieren kann, kann prima die ersten Wochen von Einführung in die Programmierung ignorieren und einfach ein Spiel coden.
- Möchtest du dein Leben automatisieren, dann geh doch an den Bash-Workshop, der jedes Semester stattfindet. Anmelden kann man sich auf thealternative.ch

Schlaf ist wichtiger, als man denkt.

- Wenn du etwas zu spät in die Vorlesung kommst und nur noch Plätze in der Mitte der Sitzreihen frei sind: Sag deinen Mitstudierenden, sie sollen auf dem Stuhl nach vorne rutschen, und klettere dann hinten über die Stühle. So muss niemand aufstehen.
- Die Ässbar ist hin und wieder vor dem CLA - dort gibt es Sandwiches von gestern zu super günstigen Preisen.
- Es ist kein Problem, seine Übungsgruppe zu wechseln. Wenn deine Gruppe nicht gut ist, wechsle unbedingt!
- Es ist tatsächlich möglich, die ETH-Drucker bei sich als lokale Drucker zu installieren, sodass man direkt aus z.B. Office drucken kann. Bonus: Auf diese Weise kommt dein Druckjob sogar dann durch, wenn das Webinterface behauptet, das System sei überlastet.
- Niemals etwas in Analysis von der Tafel abschreiben. Dabei hinkt man immer hinterher und kommt dann nicht mit. Besser, man konzentriert sich auf das, was der Dozent sagt, und versteht es dann auch.
- Benutz das Sportangebot des ASVZ. Das hilft den Kopf zu lüften. Es gibt Angebote für alle Niveaus, auch für Leute mit der Kondition eines Schwamms.
- Am 2. Oktober findet die VIS-Mitgliederversammlung statt. Das ist eine einmalige Gelegenheit, sich vom VIS bekochen zu lassen (und herauszufinden, was der VIS so macht).

Im VIS-Büro kann man coole VIS-Tassen kaufen.

- Falls dir jemand "Touch of Class" andrehen will, schlag unbedingt zu. Das Buch eignet sich wunderbar als Türstopper.
- Die ETH Bibliothek (https://library.ethz.ch) bietet sehr viele Bücher gratis zum Download an.
- Mach am Anfang der Sommer-"Ferien" eine Woche wirklich frei. Sonst drehst du in der Lernphase durch.
- Die Winter-"Ferien" sind leider etwas kürzer, aber freie Weihnachten liegen trotzdem drin.

➔

- Im ETH-Druckzentrum im HG kann man sehr günstig Poster bis Grösse A0 drucken, auch für private Zwecke. Man kann da auch Hefte binden lassen, laminieren, und so weiter.
- Noch günstiger als das ETH-Druckzentrum ist der Studentenverein SPOD (Student Print on Demand) - sie können aber nur bis A3 drucken und binden und haben nicht jeden Tag offen. SPOD befindet sich im CAB E 11.
- Mit deiner Legi hast du nicht nur Zugang zum CAB, sondern auch zum CHN und zum RZ! Im CHN hat es ziemlich gute Lernareale.
- Die Computerräume im CAB H56/57 sind rund um die Uhr geöffnet und bieten meistens genug Platz.
- Lass dich nicht von Mitstudierenden verunsichern, die vorgeben, schon alles zu können.
- Serien abgeben ist immer nützlich, auch wenn man sie nur halb gelöst hat.
- Die Events vom VIS sind ein guter Ausgleich zum Studium und du kannst Kontakte mit anderen Studierenden knüpfen.
- Als VIS-Mitglied geniesst man viele Vorteile, allen voran Gratiskaffee.
- Auch als-SOSETH Mitglied kriegt man Vorteile, zum Beispiel Gratiszugang zu virtuellen Servern oder VR-Equipment zum Pröbeln.

Die Unimensa ist günstiger als die Polymensa.

- Guter Kontakt zu den Mitstudierenden ist extrem hilfreich im Studium, sowohl zum Lernen als auch sonst.
- Es gibt Vorlesungen, bei denen Notizen nützlich sind, und solche, bei denen sie Zeitverschwendung sind. Den Unterschied zu erkennen ist der Schlüssel zum effizienten Lernen.
- Basisprüfungs-Block 1 im Winter zu schreiben ist tausendmal besser als beide Blöcke im Sommer zu versuchen!
- In der Polymensa kann man nachschöpfen. Allerdings ist der Nachschlag sehr klein und besteht nur aus der Gemüsebeilage.
- Vergiss nicht, dich für die Prüfungen anzumelden, wenn du sie machen willst. Du kannst dich immer noch abmelden, wenn du siehst, dass du es nicht schaffst.
- Im IT Shop (idesnx.ethz.ch) kann man als Studi allerlei Software gratis beziehen, zum Beispiel VMWare, Microsoft Office oder Windows 10.
- Darüber hinaus bieten viele Softwareanbieter Gratislizenzen für Studierende an. Bei JetBrains zum Beispiel kann man als Studi IntelliJ gratis bekommen.
- Als Studi kann man sich bei GitHub gratis private Repositories erstellen. Ideal, um Gruppenprojekte zu stemmen. Wer einen Aluhut trägt und mit etwas weniger Features klarkommt, kann sein Repository auch beim VIS auf GitLab hosten (gitlab.vis.ethz.ch).
- Falls du total überfordert bist und fast keine Übungen lösen kannst: Verzweifle nicht! In den Ferien lässt sich das bestimmt noch ändern ;)

- Die ETH bietet an vielen Orten (z.B. Bahnhöfen) gratis WLAN mit Internetzugang. Sobald du einen offenes WLAN-Netz findest, das Swisscom oder Monzoon heisst, kannst du dich damit verbinden, eine VPN-Verbindung zur ETH öffnen und schon hast du gratis Internetzugang.
- Die ETH ist Teil des Eduroam-Projekts. Jeder Studi einer Hochschule, die an diesem Projekt teilnimmt, erhält an anderen teilnehmenden Hochschulen gratis WLAN (also auch an der UZH oder im Ausland). Melde dich einfach im WLAN mit SSID "eduroam" an. Dein Benutzername ist "nethz_kuerzel@ethz.ch" (ohne "student"). Das Passwort ist das normale nethz-Passwort.
- Gebt euch nicht der Illusion hin, dass ihr irgendetwas begriffen habt, indem ihr die Vorlesungen besucht - Übungsblätter sind der Schlüssel.
- Falls in einer Prüfung eine selbst verfasste Zusammenfassung zugelassen ist, füllt sie nicht nur mit Theorie, sondern auch mit Musterlösungen aus alten Prüfungen. Je nach Fach kann man ganze Aufgaben 1:1 abschreiben.
- Es mag zwar verwirrend sein, aber in der realen Welt weiss nicht jeder, was Analysis ist.

ANZEIGE

NOSERENGINEERING
WE KNOW HOW

«Ich bin ein NOSER, ich lass die Kaffemaschine mit dir sprechen.»

Tobias, 32, Software Engineer

Möchtest du ein NOSER werden und die Zukunft mitgestalten?

Dann bewirb dich bei uns. Ein Ort, an dem echte Wertschätzung gelebt wird. Noser Engineering ist ein führendes Schweizer Software-Unternehmen für individuelle Lösungen. Hier entwickle ich zusammen mit über 170 Spezialisten erfolgreiche Produkte und Dienstleistungen. Basierend auf der breiten Erfahrung und anhand von neusten Technologien begleiten wir mittlere bis Grossbetriebe aller Branchen von der Idee über die Realisierung bis zum Betrieb und Unterhalt. Gemeinsam erreichen wir das digitale Kundenziel – we know how.

NOSER ENGINEERING AG WINTERTHUR I LUZERN I BERN I MÜNCHEN
WWW.NOSER.COM/JOBS I BEWERBUNG@NOSER.COM

ASL Tips

STEFAN DIETIKER – ALWAYS HERE TO HELP

The ASL project is comprehensive and demanding. As a student, you enjoy great freedom in how to approach the project, so I thought I would share some general insights based on my experience. Special thanks to Matthias Lanz for contributing his thoughts on the matter.

Disclaimer: We make no claim to completeness. Also, this article does not necessarily reflect the opinion of the HoPo commission, the VIS in general–let alone the course staff of ASL. In case of doubt, statements by the course staff always take precedence over what is stated in this article. It cannot act as a reference point. Furthermore, the course contents and structure are subject to change and therefore this article might not be up to date.

General Remarks

It Is Time-Consuming

If ASL is part of your personal study programme, it is safe to say that the ASL project will be one of the most comprehensive projects that you will have completed by the end of your master's studies here at ETH. It also safe to say that it will take up a considerable amount of your time during the semester. Be sure to reserve enough time.

Cleanliness & Completeness Over Beauty

In general, the graphs and plots that you provide must be complete. If you are unsure what to include in what plots (e.g. error bars), ask the assistants. However, do not spend too much time on making your plots and graphs nice-looking. In general, the assistants know what to expect, and bells and whistles do not count. What is important is that they are complete and readable.

Discuss your Solutions

Try to explain your approach to colleagues. In the process of explaining your (conceptual) solution to someone else, you will test and potentially improve your understanding of your own system. However, discussing a solution is not the same as copying it! Do not even think about copying code or data from colleagues. You will not be able to understand what is going on.

There is, of course, the danger of spreading falsehoods among students. Be sure to consult the course staff in case of questions concerning requirements, relevant definitions, etc. They are the ultimate authority.

It happened a number of times that while discussing my solution with someone else, we found ambiguities in our underlying assumptions which we then clarified with our individual assistants.

Implementation

Don't Try to Be a Smart-Ass

It sounds ridiculous, but it is best not to try to be smart about how your solution is implemented. In the end, nobody cares about absolute performance values. What the assistants–and therefore you–care about is what is going on in your system.

Been there, done that. When I took the course last year, I first implemented a version with zero-copy buffers–only to find out that I introduced a bottleneck in one part of the system where multiple threads competed for a lock. Since I am a masochist, I reimplemented the system using a simpler approach and–guess what–it performed better than the first version.

Test-Driven Development

The following applies not only to ASL, of course. But it is very useful to employ test-driven development (TDD). In particular, since the development platform is Java and there is good support for unit testing. I can only restate the benefits described elsewhere: it is not worth unit testing every little function, but TDD will help you write correct code faster. Also, the interfaces of your classes tend to be cleaner, and you tend not to resist necessary refactorings.

Deploy as Soon as Possible

This cannot be overstated: Deploy as early as possible and test your system in the cloud. First of all, you are probably underestimating the time it takes to get the system to run–even if you are sure that you have set aside enough time, still behave as if you did not.

Secondly, you might encounter nasty bugs in your code: In my original system, I had a subtle bug that only surfaced when the network latencies were above a certain level. As a result, when testing locally, everything seemed to be going well. Only after deploying the system to the testing environment, for the first time, the system started producing spurious results.

Experimentation and System Setup

Jumphost

Set up a (cheap) additional virtual machine and use it as a jumphost. You can run the scripts controlling the execution of the experiments (logging into the machines, starting the clients, collecting the logs, etc.) on the jumphost.

Automation

One could argue that this is the essence of software engineering anyway, but automate as much as possible. That is, carefully consider what configuration parameters your experiments require and log them alongside the results. You will be conducting a great number of experiments and traceability is key.

You enjoy almost complete freedom in what language to use for controlling the experiments and analysing the data. You will not get around writing some UNIX shell scripts. However, my recommendation is that you use a single language to cover as wide of a range of the different tasks that you are facing. Python is a good candidate in that respect, as its standard library provides good support for standard scripting tasks (file ➔

system manipulation, starting/stopping processes, etc.). However, there are also powerful libraries for statistical analysis and plotting, such as numpy [1] and matplotlib [2].

Noise Reduction

Keep in mind that your experiment runs on infrastructure that is shared with others. Try to run the experiments at night and outside of business hours of the time zone the data center is located in. This should reduce the noise level and make individual runs more comparable.

Modelling

Don't Compare Reports

In light of the remarks above about discussing solutions, it might be useful to discuss the application of models with your colleagues to test your understanding of them. However, it is futile to compare your numbers and plots to someone else's. Both the system itself, the experimental configuration that acts as a basis for your analysis, and thus also the model parameters might differ substantially. It is essential that you understand what you are doing and are able to provide a coherent and consistent explanation of your choices in the report.

Backup Your Claims

Backup each of your assumptions with a reference to either lecture slides or literature. If you make wrong assumptions and/or cannot justify your decisions, entire experiments or analytical results might be considered invalid.

References

[1] http://www.numpy.org/

[2] http://matplotlib.org/

From ETH Spin-off to global travel brand

Seven years ago, six young startup founders met right here at ETH.

After graduation, they built GetYourGuide: a travel activities platform that helps you find and book fun things to do in 2.500+ destinations worldwide.

With over 300 employees and head offices in Zurich and Berlin, GetYourGuide is one of Switzerland's biggest startup success stories to date.

Come join the adventure as an engineer. Work on small, agile teams. Build things fast. Constant brainstorming, constant iteration, constant improvement.

Join our Zurich team

ethz.getyourguide.com

Reisebericht: Zernez – Trento

ZENO KOLLER – JUST RIDING ALONG

Da ich in der Woche vor der Artikel-Deadline auf einer Velotour war, schrieb ich konsequenterweise einen Artikel darüber.

Tag minus 1: Planung

Technisch gesehen war die Tour eine Bieridee. In einer Beiz schlug mein Kommilitone David vor, das Semesterende mit einer Radtour zu feiern. Inspiriert vom Vorschlag und eigenen nostalgischen Erinnerungen an vergangene Veloferien schlug ich ein. Die Route war schnell gefunden: Vom Münstertal durchs Südtirol ins Trentino. Das erfüllte unsere Kriterien: Alpenpässe, höchstwahrscheinlich gutes Wetter und nicht allzu teuer. Zurück würden wir dann den Zug nehmen. Auf dem Hinweg wählten wir als fahrbaren Untersatz das Rennrad. Damit kommen wir schnell voran, müssen aber unser Gepäck im Rucksack tragen. Die Packliste enthält also nur das allernötigste in der leichtesten Ausführung.

Tag 1: Zernez – Santa Maria

Da in Italien am geplanten Rückreisetag die Bahnangestellten streiken, fahren wir einen Tag früher ab. Die Rhätische Bahn bringt uns nach Zernez. Das ist der Startort unseres «Warm-Up»-Tages, an dem wir den Ofenpass überqueren, um ins Münstertal zu gelangen. Die Passstrasse führt durch den Schweizer Nationalpark. Eine ungewohnte Landschaft. Bei so viel Wald fühlt man sich eher in Kanada als in der verhäuselten Schweiz! Der Pass wird seinem Namen in zweierlei Hinsicht gerecht. Erstens brennt die Sonne, zweitens wimmelt es von Motorradfahrern – es ist Sonntag. Unzählige Öfen überholen uns auf der zum Glück grosszügig angelegten Strasse. Nach einem verdienten Gipfelsandwich rasen wir ins Val Müstair runter.

Aussicht von Lu

Wir sind so früh in Santa Maria, dass wir nicht gleich in die Jugendherberge einchecken können. So erkunden wir das Dorf. Die Häuser sind alle kunstvoll verziert, auch die Jugi ist keine Ausnahme. Obwohl es in Santa Maria allerlei Restaurants gibt, packen wir nach dem Einchecken und der verdienten Dusche die Räder auf das letzte Postauto ins 500 Meter höher gelegene Lü. Im hoch über der Ofenpassstrasse gelegenen Ort speisen wir als einzige Gäste im Restaurant Tschierf Capuns und trinken ein feines Engadiner Weizen. Der Ausflug hat sich gelohnt, hat Lü doch deutlich mehr Abendsonne als der Talboden. Die kurvenreiche Abfahrt ist das Dessert.

Tag 2: Santa Maria – Umbrail – Stelvio – Meran

Bei der Dorfbäckerei decken wir uns mit Proviant für den Tag ein. Unter anderem eine Bündner Nusstorte – Kassensturz-Testsieger! Mehr dazu später. Nach dem Frühstück gehts gleich zur Sache. Wie gestern starten wir senkrecht. Mitten in Santa Maria beginnt die Strasse auf den Umbrailpass. Da wir heute etwas höher hinaus wollen, gehen wir es gemütlich an. Kehre um Kehre im schattigen Nadelwald. Heute Montag hat's nicht mehr so viel Verkehr wie gestern. Dafür kommt er schubweise: Ein Porscheklub, eine Gruppe von KTM-Ultraleicht-Cabrios und rund 10 Old School Vespas inklusive Gepäck rasen, zischen, beziehungsweise knattern an uns vorbei.

Auch das Zollgebäude an der Grenze zu Italien ist so schön wie die Häuser im Tal. Es ist nicht viel los, und wir nutzen die Ruhe für eine Pause, den 250 m höher gelegenen Stelviopass (Deutsch Stilfserjoch) im Blick. Darunter Schneefelder. Bei der Weiterfahrt ist es dann vorbei mit der Ruhe: Gümmeler und Töfffahrer begleiten uns auf den letzten paar Kehren. Auf der Passhöhe geht es zu wie auf einer Chilbi. Souvenirs, Süssigkeiten und gleich drei Wurststände. Umso eher nehmen wir die Abfahrt unter die Räder. Uns stehen 50 Spitzkehren und 2000 Tiefenmeter bevor. Die Bremsfinger erholen ➔

Der höchste Punkt

wir in der Mitte der Abfahrt beim Mittagessen. Unten angekommen, befinden wir uns im Vinschgau. Immer noch Italien, aber hier spricht man Deutsch. Vinschgau in einem Wort: grün. Endlos scheinen die Äpfelplantagen, durch die der Veloweg führt. Es ist heiss und wir kommen bei freier Fahrt schnell voran. Die gut angelegten Radwege ziehen aber auch ein entsprechendes Publikum an. Es wimmelt von älteren Radtouristen, die fast ausschliesslich talabwärts unterwegs sind. Sicher eine gute Investition in den Tourismus!

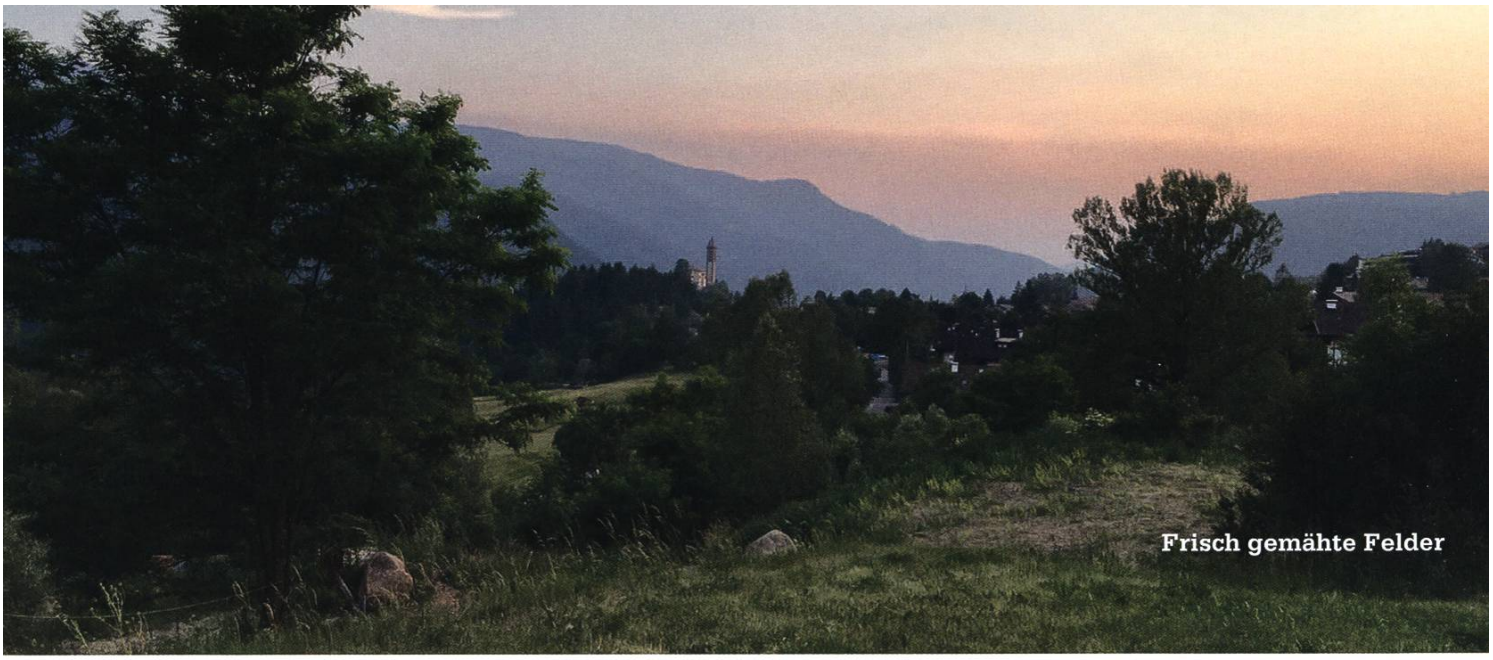
Durch das ständige Beschleunigen und Verzögern in der Hitze und dem Aufstieg am Morgen sind wir bald so geschlaucht, dass wir auf den 60 Kilometern nach Meran nochmals eine Pause machen. Nun kommt sie, die Nusstorte. Ein Trip auf Karamell und Baumnüssen. Fantastisch! Als wir am späten Nachmittag in Meran ankommen, ist es mindestens 35 Grad. Wir steuern eine Jugendherberge an, die ist aber schon ausgebucht. Nach so einem Tag eine Bleibe zu suchen, gibt uns den Rest. Mit einer Prise Glück kriegen wir noch das letzte Hotelzimmer und verspeisen je eine gigantische Pizza. Eine Herausforderung, nach diesen Strapazen.

Tag 3: Meran – Bozen – Auer – Val di Fiemme

Zum Start in den Tag gibt's ein italienisches Frühstück. Wer es nicht kennt: Das Brot ist im Gegensatz zum Schweizer Frühstück deprimierend. Dessen Qualität wird jeweils grosszügig mit einem Kuchenbüffet kompensiert. Gut gemeint, aber nicht kompatibel mit unserem Geschmack. Löblich ist dafür der lokale Brie, der frisch gepresste Orangensaft und das Joghurt, welches wie gestern aus der Region stammt.

Was als Entspannungs-Etappe geplant war, stellt sich als der härteste Tag bis jetzt heraus. Zuerst bolzen wir nach Bozen. Der Radweg ist so perfekt angelegt wie gewohnt. Auf einer Seite liegt jeweils der Fluss Etsch, auf der anderen die bekannten Äpfelfelder. Wie am Vortag zuvor fädeln wir uns durch die Reiseradler. Die Kilometersteine am Wegrand machen es zum Kinderspiel, sich in 3-Kilometer-Ablösungen den Windschatten zu teilen. Kurios, aber sinnvoll dünkt uns der Selecta-Automat mit Erfrischungen für Velofahrer am Wegrand. Ab Bozen gibts zu den Äpfeln immer mehr Trauben. Tramin ist in der Nähe.

Montagna, Südtirol


Frisch gemähte Felder

Nach dem Flachstück hört der Spass auf. 1000 Höhenmeter Aufstieg ins Val di Fiemme bei humorlosen 40 Grad. Es ist so heiss, das der Teer an den Reifen kleben bleibt. Salzflecken zeichnen sich am Rucksack ab. Mit immer kürzer aufeinander folgenden Pausen kämpfen wir uns die Wand hoch.

In Truden, dem letzten offenbar deutschsprachigen Ort, haben wir es geschafft. Wir lösen die bitter nötige Siesta ein und essen im Schatten eines Zwetschenbaumes zu Mittag.

Ein Schläfchen, ein Käffchen und rund 10 Kilometer später kommen wir in Castello di Fiemme an, dem Etappenziel. Hier spricht man definitiv Italienisch! Unsere Absteige ist nicht schlecht. Eine geräumige Airbnb-Wohnung im Dachgeschoss eines Bienenzüchters. Durch die Erfahrung vom Vorabend haben wir vorgesorgt und am gleichen Abend gleich die restlichen Übernachtungen organisiert.

In Castello hat es nicht viele Restaurants. Müde wie wir sind, nehmen wir den Ortsbus hoch nach Cavalese, den nächsten grösseren Ort. Wir finden ein gutes Restaurant, welches lokale Spezialitäten anbietet. Mit vollen Bäuchen laufen wir auf Feldwegen zurück nach Castello. Die frisch gemähten Felder duften nach Heu an diesem lauen Abend.

Tag 4: Castello di Fiemme – Pergine

Wir starten etwas früher als gestern, um der Mittagshitze zu entgehen. Da es etwas bedeckt ist, ist es nicht so heiss wie am Vortag. Die vierte und letzte Etappe führt durch Täler des Trentiner Hinterlands. Die Strasse windet sich durch dunkle Tannenwälder. Ab und zu sieht man Türme von Baumstämmen. Mehr Twin Peaks als Italien! Bei einem Zwischenstopp flicken wir den ersten und letzten Plattfuss der Tour. Dieser ist selbst verschuldet. Nach dem Aufpumpen eines weichen Hinterpneus ging das Ventil kaputt. Neuer Schlauch! Rauf, runter, rauf gehts weiter durch Wälder. Der Passo del Redebus ist der letzte längere Aufstieg. Danach kommt die zweifellos beste Abfahrt der Tour. 800 Höhenmeter, fast kein Verkehr und lange Geraden. Die surreale Stimmung trägt das Ihre dazu bei. Im Kontrast zu tiefblauen, von dunstgetränkten Schwaden umhüllten Hügeln steht die rostrote Strasse, umgeben von grellgrünen Bäumen. Die Müdigkeit wirkt darauf wie ein Instagram-Filter. Am Lago di Caldenazzo lunchen wir und hängen noch eine Weile am See rum.

Dann geht es nochmals hoch zum Schloss, das auf einem Hügel über Pergine liegt. Auf das Schloss, welches eigentlich ein Hotel und Restaurant ist, sind wir durch den Hinweis von Davids Vater gelangt.

➔

Castello Pergine

Die Wirte sind seit 25 Jahren ein charismatisches Schweizer Ehepaar. Die meisten Gäste, so erfahren wir in Gesprächen, kommen schon seit Jahren immer wieder hierher. Für die meisten ist es das letzte Mal. Das Wirtepaar plant, den Betrieb im Frühjahr 2018 zu verkaufen.

Warum die Gäste immer wieder hierher kommen? Weil man eben im Schoss wohnt. Das beginnt bei den gut renovierten, aber stilistisch passenden Zimmern und Aufenthaltsräumen. Dazu kommt der Schlossrasen, der zum Ausruhen einlädt, müde vom Bestaunen der Aussicht. Das fürstliche Nachtessen rundet die Schloss-Erfahrung ab. In der Halbpension inbegriffen ist ein exquisites Mehrgangmenu, dazu wird erstklassiger Wein aus der Region serviert. Hat man davon noch nicht genug, wird einem bei der Bar weitergeholfen…

Tag 5: Rückfahrt

Heute steht leider schon die Heimreise an. Das Frühstück im Rittersaal ist dem Abendessen ebenbürtig. Allzu lange dürfen wir nicht verweilen, wir müssen 15 Kilometer nach Trento fahren. Zum Abschied drückt uns die Schlossherrin noch ein Stück Kuchen für die Heimreise in die Hand. Mithilfe vom iPhone an Davids Lenker finden wir den Weg durch den wilden italienischen Morgenverkehr, abseits der Tunnels, dessen Benutzung in Italien Radfahrern teilweise untersagt ist. Unser Zug fährt über den Brenner nach Innsbruck. Dann geht es weiter Richtung Feldkirch, nahe der Schweizer Grenze. Da die Zugstrecke nach Sargans durch Bauarbeiten unterbrochen ist und wir mit den Fahrrädern keine Lust auf den Bahnersatzbus haben, wird das zur N+1-ten Etappe. Here we go again! Kurz gesagt: Drei Länder in weniger als einer Stunde. Der Zollbeamte am Übergang nach Liechtenstein hat nicht mal ein müdes Lächeln für uns übrig. Nochmals Windschatten an einem Fluss entlang, diesmal dem Rhein. Dass wir beide richtig Gas geben, ist nötig. Wir gelangen auf den Bahnsteig, als gerade der um drei Minuten verspätete Zug nach Zürich einfährt. Zum Dosenbier, welches wir vorher schon noch kaufen mussten (ähem), essen wir den Schlosskuchen. So fährt auch das Bier später ein…

Fazit der Tour: 307 km tönt zwar nach nicht allzu viel, wir waren aber beide wegen des Semesters nicht ernsthaft trainiert. Die 6200 Höhenmeter sind kein schlechter Benchmark. Die Packliste hat ausgereicht. Fehl am Platz war nicht mal die Regenjacke – zusammengerollt das perfekte Kopfkissen. Wir sind uns einig: Das wird nicht unsere letzte solche Radtour sein

ESF 17

Digital Adventures

28/09/2017
20:00 - 03:00
ETH Hönggerberg

Free Shuttle Service
Free Entrance for Erstis

LIGHT & SOUND
HOLY COW!
Burrito
Zürcher Kantonalbank
vseth Verband der Studierenden an der ETH

Build Your Own Multimedia Center

ALINE ABLER – WIELDS TECHNOLOGY LIKE MAGIC (SOMETIMES)

My dad is like me – he likes fancy new things. He doesn't tinker with them like I do, though. But when he got himself one of these huge Ultra-HD curved TV screens, I got his old TV, so I'm not complaining.

So I suddenly found myself in the situation of owning a TV, but no TV subscription, because I only recently switched to an internet-only glass fibre plan. Not that this is a problem – my PC connects to the TV just fine, and that's enough for me. Cable isn't what it used to be, anyway.

Homebrew Cinema

But naturally, that's not all there is to it, otherwise I wouldn't be writing this article. The problem is that I'm a lazy person. When I want to watch a movie, I don't want to go find the remote in order to turn on the TV. No, it would be much more convenient if I could turn on the TV screen from my PC – which is usually running, anyway. Just imagine this – a single command, and click, your TV turns on, your PC screens turn off, your lights dim, and ta-dah, cinema mode activated.

However, that was much easier said than done. It's surprisingly hard to communicate with a screen. I quickly found out that there actually exists a communication protocol for HDMI connections – it's called CEC, or Consumer Electronics Control. The name already says it – "consumer electronics". The original idea behind CEC was that you could use one remote for all your devices. When you turn on your TV, your DVD player or satellite receiver should turn on as well. Your TV is then supposed to "forward" all the remote commands to the other devices, which then act accordingly – so your DVD player can react to a press of the "rewind" button on the TV remote. The idea is neat, but then all the consumer electronics manufacturers did their own fancy things to make sure their products would become as incompatible as possible with other manufacturer's products, and in the end, CEC never became what it was supposed to be.

It's surprisingly hard to communicate with a screen.

But still, there was a communication protocol suitable for what I wanted to do, and that's a huge step forward. I quickly found out that my TV does indeed support CEC – or, as Samsung calls it, "Samsung Anynet" – but my PC doesn't.

That's right; computers are apparently not "consumer electronics". CEC actually requires hardware support, and my research showed that PC manufacturers in general do not bother with CEC, so I had no way of using it.

That sure sounds like I was out of options, but through my research, I also found something that could help me – an external CEC adapter. It seems like there is precisely one company on the globe that produces a so-called USB-CEC adapter[1]. This nifty little device has two HDMI ports and a USB port. One HDMI connection connects to the TV, while the other HDMI connection as well as the USB one attach to the PC. The adapter taps the CEC signal and converts it to a USB signal, which is understood by your PC through a custom driver. The actual picture signal is not altered at all, so there is no effect on image quality. Neat.

It seems like there is precisely one company on the globe that produces these things.

The adapter costs 40 bucks, but after reading a few reviews, I decided to go for it. I did not regret it. I found out that with this little thing (it really is tiny, you'd be surprised), I could do so much more than just turn on my TV from a bash script. I mentioned before that CEC is also meant to forward signals from your remote control, and now I could read out these signals on my PC. I can now use my TV remote to control my PC! ➔

ANZEIGE

Gut, dass Verena Kuhr so beweglich ist. Das kommt ihr nicht nur als «Mariechen» im Karneval zugute. Akrobatisches Geschick zeigt sie auch als Software Engineer bei BSI. Denn es erfordert viel Flexibilität, um sich mit 250 Kollegen an 7 Standorten so aufzustellen, dass jeden Tag eine perfekte Performance entsteht. Um so schöner ist der Applaus der Kunden, welche diese Beweglichkeit sehr schätzen.

Verstärke unser Team (auch Praktikum möglich): **www.bsi-software.com/kollegen**

BSI Business Systems Integration AG

bsi

I quickly threw together a bash script that forwards the CEC signals to my favourite media player, mpv. I can only recommend mpv – it plays anything and everything, including youtube videos and a variety of online video streaming sites. And now, with my little script, I could pause and seek my videos from across the room using my TV remote. My favourite use case is watching movies from my bed – when the movie is over, I can have my bash script automatically turn off the TV and shut down my computer, and, since I have smart lights[2], it can turn these off as well. Fall asleep while watching? No problemo!

My favourite use case is watching movies from my bed.

At this point, I would like to note that while most PCs do not support CEC, the Raspberry Pi actually does. Many people use the Pi as a cheap media center, usually with the Kodi suite installed. Kodi actually comes with a plugin for CEC, so if you do have such a Pi-powered media center, chances are you can use your TV remote to control Kodi. You can also turn on the TV from your Pi. A friend of mine, Sandro, has his Raspberry connected to a beamer, which also supports CEC, so this solution works in his setting too (although, of course, Sandro's home cinema setup is still way inferior to mine, especially because mine was first and he stole all my ideas). This shows how widely used CEC actually is - once you know about it, there's a lot that can be done.

Audiophiles will yell at me now.

Streaming Fun

But of course, I don't spend all of my free time watching TV. Sometimes I just listen to music. And because my PC speakers are total crap, I eventually got into the habit of turning on my TV screen to play my music from the TV's speakers. Which, of course, is total overkill. And the TV speakers aren't even that good, either.

So I picked up my old compact stereo system which had been collecting dust at my parents' house for a while now. It's a rather old-fashioned thing. It plays radio and CDs, but it doesn't have any fancy Bluetooth capabilities. However, it sports a single AUX input, which would do the job for me.

Audiophiles will yell at me now, but I just went with the resources I had. I hooked up my stereo system to my Raspberry with an AUX cable, and with that setup, I could produce better sound than with my TV, so it was definitely a step in the right direction. The issue was just that my music resided on my PC, not my Raspberry Pi.

But I already had something in mind: UPnP streaming. UPnP is for "Universal Plug and Play", and that pretty much sums it up. It's a set of networking protocols that allow for devices to automatically discover each other and share data.

To put this in simpler, less accurate terms: With UPnP, you can stream content via your local network.

The concept is simple and surprisingly versatile: In your local network, you can have media servers, media controllers and media renderers. The server just stores media – that would be the PC in my case, but a NAS would also suit the role. The renderer plays content from a server, but it doesn't do anything on its own – it needs a controller, which sends instructions about which file from which server it should play. In a classic textbook setup, you would have a NAS, a TV and a smartphone interact in this way. The NAS naturally stores the media, so it is a server. The TV plays media, so it has to be a renderer. The smartphone does neither – it is simply used to control which media is playing, like a remote control. The smartphone acts as the controller.

The concept is simple and surprisingly versatile.

My setup is a bit less out-of-the-book, but it too consists of one server, one controller and one renderer. The PC takes the role of both controller and server, while my Raspberry Pi acts as the renderer.

I quickly found suitable renderer software. I opted for gmrender-resurrect[3], a relaunch of the abandoned GMediaRender project. It does precisely what I need it to – plus, it's lightweight, uses few resources, and runs headless.

Finding a suitable server and controller was tougher, although there's definitely no shortage of UPnP server software. Wikipedia is a great resource for finding them, by the way[4]. My issue was that none of the controllers really suited my needs – I wanted a command line interface, but I only found GUI programs. Oh, what a terrible pointy-clicky world we live in!

But I eventually found a solution in the form of pulseaudio-dlna[5][6]. For the uninitiated: PulseAudio is the most widely used Linux sound system, and it's in charge of mapping your sound sources (i.e. different applications producing sound) to your various sound sinks (different speakers, for example). Pulseaudio-dlna now acts as an interface between my local PulseAudio instance and all the UPnP renderers on my network, by simply adding them as sound sinks to my system. In other words: My Raspberry Pi now shows up as a "virtual speaker" on my system. This lets me stream the sound of any application live (well, with a two-second delay) to my Pi, which plays it on the stereo system. Works like a charm!

With this setup, I could now use any old music player to play my music, so naturally, I chose mpd[7]. That's short for "Music Player Daemon", and as the name suggests, this player runs in the background – no GUI involved. In fact, mpd follows a client-server model – mpd indexes and plays your music, while some mpd client tells mpd what music it should play. This is actually very similar to the UPnP concept – mpd acts as server and renderer, and the client is the controller.

As it turns out, mpd can be used in a multitude of ways. In its extensive configuration file, I found a way to define multiple sound outputs. I decided to add three: my local PC audio, my TV's speakers (accessed via HDMI) and, of course, my Raspberry-powered UPnP renderer. All three of these were present on my system as PulseAudio sinks, so adding them to mpd was easy – and I could now enable and disable them individually. ➔

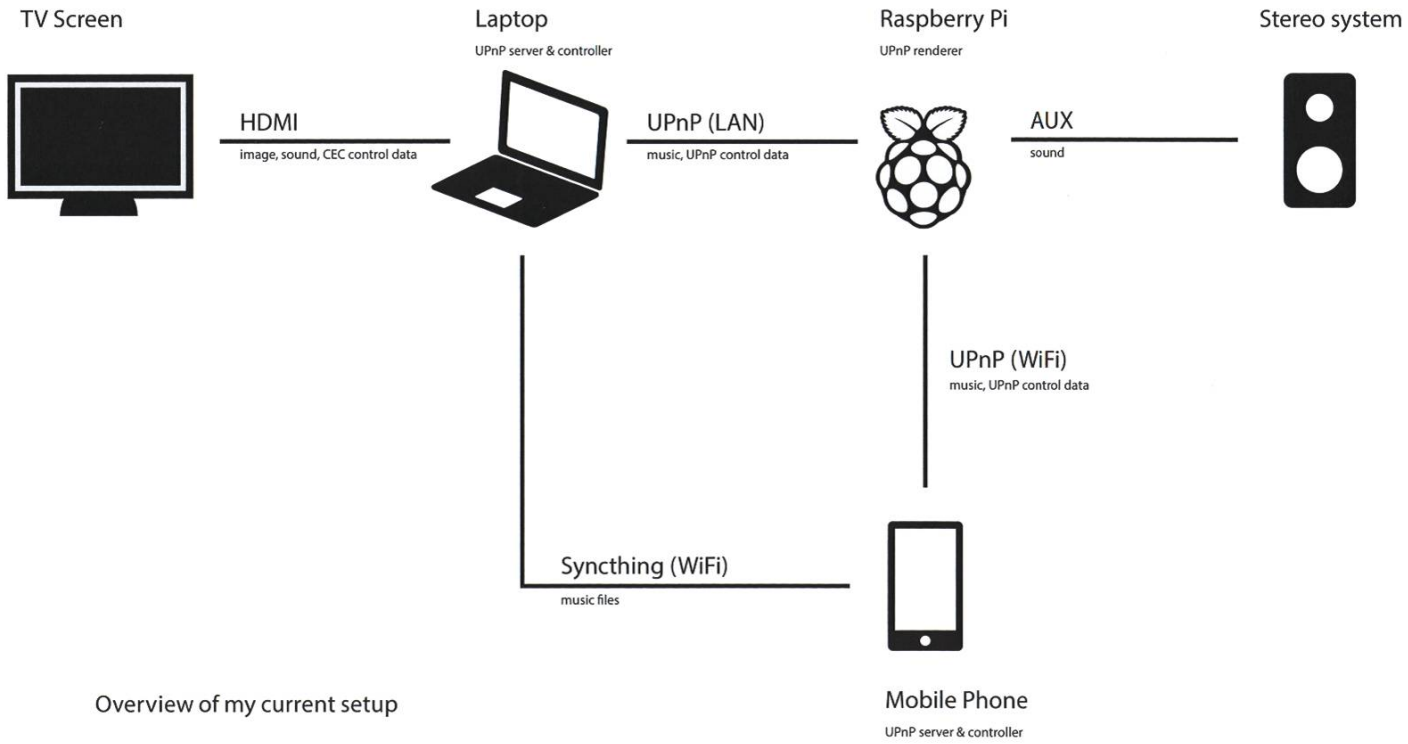
There are some pretty fancy GUI clients for mpd, but I just picked mpc, a minimalistic command line interface. You might be wondering why I am so obsessed with the command line. Well, in this case, the reason is simple: I want to control music playback entirely through keyboard shortcuts, and you can't define keyboard shortcuts for GUI program actions.

So, finally, I was all set. I repurposed my numpad to work as a music controller, with keys to play, pause, skip, shuffle and change audio output. Perfect!

But I still wasn't done.

But I still wasn't done. While I do keep my PC running most of the time, there are still rare occasions where I want music, but my PC is off. And since my PC does all the music streaming, that's a bit of an issue.

Fortunately, I have other devices which I keep running 24/7 and also contain all of my music – such as my phone. A while ago, I had set up Syncthing[8][9] to synchronize my entire music library between my phone and PC. Originally, I did that so I could listen to my music on the go, but now it came in handy for my little problem. I just needed to find an app to turn my phone into a UPnP server and controller.



Overview of my current setup

There are quite a few such apps, and I've tried way too many, but eventually, I stuck with BubbleUPnP[10]. It is, unfortunately, "freemium" – the free version doesn't support playlists with more than 15 tracks, among other things – but it was the only one that did what I wanted and worked reliably. In any case, I can now easily play music from my phone on my stereo system – thanks to my Raspberry Pi and UPnP.

And On It Goes

This is where I am now, but I still have plans to further improve my self-made cinema. For example, I still can't play a movie's sound on my stereo system, because pulseaudio-dlna introduces that pesky two-second delay. And the Pi's audio output just isn't very good, so maybe I'll get an audio card some day. Or an infrared transmitter, so I can control my stereo system from the PC as well. I'll see.

I hope I could encourage you to tinker with your own setup a little. I've showcased a few concepts that are very versatile and can be useful in most settings, so just go ahead and try them! It's fun, and you'll learn something!

References

[1] https://www.pulse-eight.com/p/104/usb-hdmi-cec-adapter

[2] See also my article on smart lights in VISIONEN 3/17

[3] https://github.com/hzeller/gmrender-resurrect

[4] https://en.wikipedia.org/wiki/List_of_UPnP_AV_media_servers_and_clients

[5] DLNA is another standard that incorporates UPnP. In the community, both acronyms are often used interchangeably.

[6] https://github.com/masmu/pulseaudio-dlna

[7] https://www.musicpd.org/

[8] https://syncthing.net/

[9] See also Sandro's Article on Syncthing in this VISIONEN

[10] https://play.google.com/store/apps/details?id=com.bubblesoft.android.bubbleupnp&hl=en

Syncthing

SANDRO KALBERMATTER – HAT RASPBERRITIS IM ENDSTADIUM

Sandro studiert Informatik im ersten Mastersemester. Sein Bachelorstudium hat ihn mit einer ordentlichen Portion Paranoia versorgt, weswegen er seine Daten jetzt wild in der Schweiz herumrepliziert.

<hype>

Du hast soeben dieses unglaubliche Foto auf deinem Handy geschossen. Das eine Foto, wofür du gelebt hast, wozu du geboren wurdest. Und eine Sekunde darauf rutscht dir dein Telefon aus den schweissnassen Fingern. Wie in Zeitlupe kriegst du mit, wie sich das Gerät unweigerlich dem Boden nähert, diesen aber nicht erreicht, weil es weiter draussen als das Brückengeländer ist. Während dein Gesicht sich schmerzvoll verzieht und dein Mund sich zu einem «oooooaaaaaaaaahhhhhhhhhnnnneeeeeeeiiiiiiiin!!!» öffnet, verschwindet das schwarze Kästchen aus deinem Sichtfeld und du hörst, wie es mit einem satten Plumpsen ins Wasser fällt. Was denkst du? Trauerst du schon um dein Foto? Nein, denn dein Handy ist wasserfest und du benutzt Syncthing, das Programm, das alle Probleme der Menschheit mit einem Schlag löst und den Messias auf Erden bringt. Während dein Handy also in der Tiefe verschwindet, hat es noch genau solange Netz, dass deine treue Syncthing-App dein unglaubliches, atemberaubendes Foto fertig auf deinen Server lädt, bis der Wasserdruck dein Handy zu Tode presst. Und das beste daran: Die NSA hat dein Bild nicht, denn es ist nie unverschlüsselt auf einem Gerät gelandet, das nicht dir selber gehört.

Trauerst du schon um dein Foto?

Dramatik beiseite. Syncthing ist ein dezentraler Synchronisierungsdienst, der beliebige Daten verschlüsselt von Host zu Host überträgt. Dezentral bedeutet, dass es nicht eine Maschine geben muss, die ständig läuft, sondern dass sich alle verbundenen Geräte, die einen Ordner teilen, untereinander synchronisieren. Dabei kommt ein BitTorrent-ähnliches Protokoll zum Einsatz (genannt Block Exchange Protocol bzw. BEP), welches die Daten in Stücke aufteilt und diese zwischen mehreren Maschinen gleichzeitig hin- und herschicken kann. Das heisst, dass du, wenn du 10 Freunde hast, die allesamt über einen traurigen Upload von 1 MB/s verfügen (mein Beileid), trotzdem mit bis zu 10 MB/s

saugen kannst, sogar eine einzige Datei. Denn Syncthing kann von jedem Host ein anderes Stückchen anfordern und so dieselbe Datei von N Peers gleichzeitig beziehen. Da Syncthing nur auf Geräten läuft, denen du vertraust und jeglicher Traffic state-of-the-art-verschlüsselt ist, ist die Sicherheit gewährleistet. Und das Beste an Syncthing: Es ist Free and Open Source Software (FOSS), das heisst, dass jeder den Quellcode ansehen, verändern und verteilen darf (und natürlich ist die Benutzung des Programms gratis). Dir fehlt ein Feature? Bau es rein!

FOSS rules!

Wer kennt noch BitTorrent Sync (heute heisst es Resilio Sync)? Es ist sozusagen der Vorgänger von Syncthing. War mal ein sehr beliebtes Programm, aber heute schwindet die Popularität: Wenn du die Auswahl zwischen Blechen und Limitierungen hättest, wie schnell würdest du dann zu Syncthing wechseln? Eben, genau. FOSS rules! In der 456. Episode von Security Now! hat sogar Steve Gibson Syncthing als potentiellen Ersatz für BitTorrent Sync vorgeschlagen [1].

Einrichtung

Unter Linux benutzt man den Paketmanager, unter Windows installiert man ein Setup (ich empfehle SyncTrayzor, welches Syncthing wrappt und dabei sehr gute Arbeit leistet), auf einem Mac nutzt man Homebrew und unter Android holt man die Syncthing-App aus dem Play Store oder F-Droid, einem Open-Source-Alternativ-App-Store. In allen Fällen kann man auch die GitHub-Repo selber klonen und kompilieren, wenn man die Bleeding Edge braucht.

Um zwei Syncthing-Hosts (X und Y) miteinander zu verbinden, startet man Syncthing, geht in den Browser auf Gerät X unter http://127.0.0.1:8384/ und guckt sich unter «Aktionen->Eigene Kennung» die ID des Geräts X an (ein strichgetrennter String aus zufälligen Zahlen und Zeichen). Diesen kann man auf dem anderen Gerät Y eingeben oder den darunter befindlichen QR-Code scannen. Gerät Y versucht dann die Kontaktaufnahme. Sobald sich X und Y sehen, meldet X, dass Gerät Y aufgenommen werden möchte. Ein Klick in der Web-GUI und die beiden kennen sich.

Nun können Ordner geteilt werden. Ordner haben einen globalen Namen und eine optionale lokale Kennung sowie natürlich einen Dateipfad und Einstellungen (dazu später mehr). Danach kann unten angekreuzt werden, mit welchen der bekanten Geräte der Ordner geteilt werden soll. Ein Klick auf «Speichern» und auf allen ausgewählten Geräten erscheint (früher oder später) eine Meldung, dass Host H den Ordner O teilen möchte. Nimmt man die Anfrage auf einem der angekreuzten Geräte an, landet man abermals im Menü für einen neuen Ordner, nur ist der globale Name schon ausgefüllt und der Haken für das Quellgerät schon gesetzt. Beliebige Ordner können mit beliebigen Geräten geteilt werden, doch muss man darauf achten, dass das Teilen zweiseitig ist: Wenn mein Handy einen Ordner mit meinem Laptop teilt, so muss mein Laptop denselben Ordner ebenfalls mit meinem Handy teilen. Ist dies nicht der Fall, erscheint auf dem Gerät, welches den Ordner nicht mit dem anderen teilt, eine Benachrichtigung («H möchte O teilen»).

➔

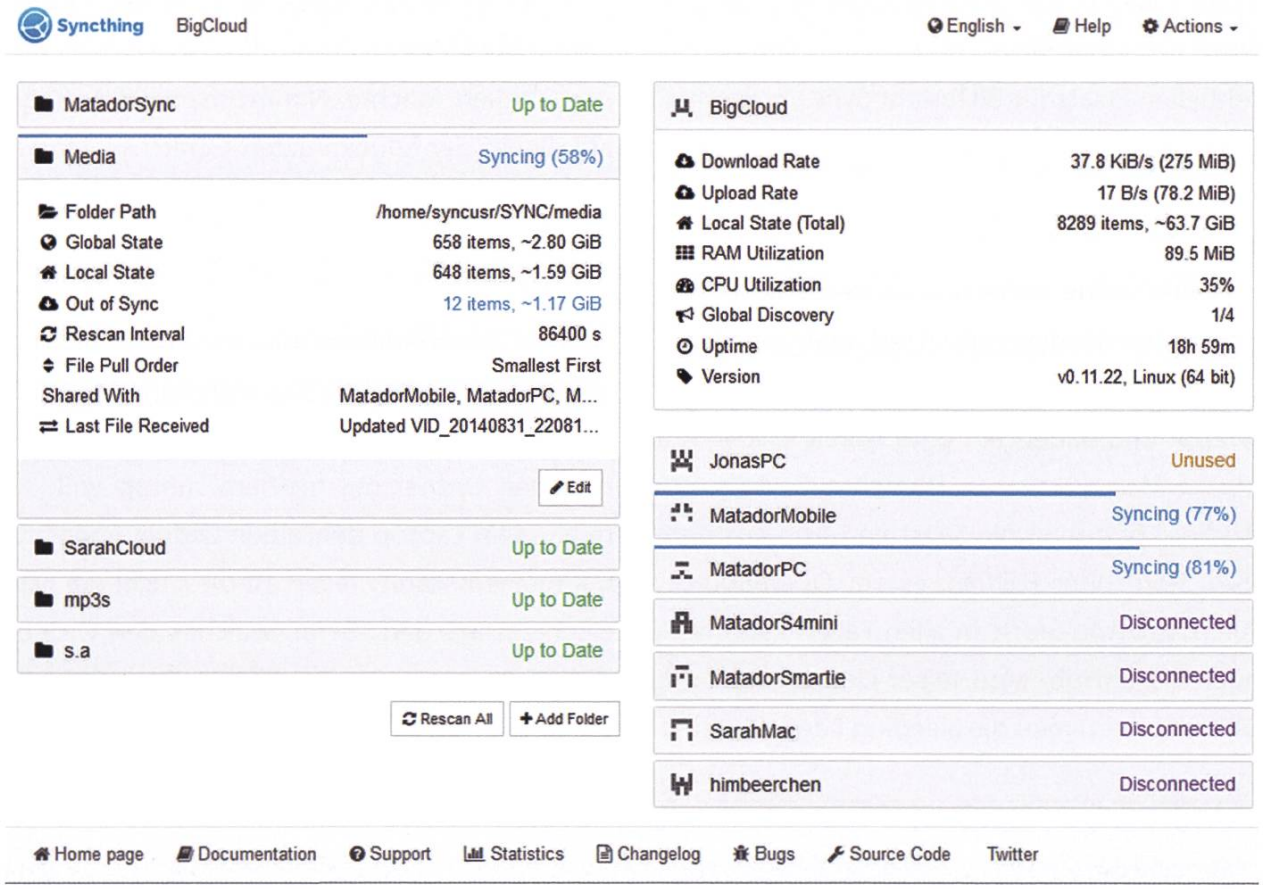
Ein paar coole Features

Es ist möglich, eine Synchronisierungskette einzurichten: Wenn mein Handy auf meinen Laptop synchronisiert, der auf den Laptop meines Freundes Bob synchronisiert, so kennen sich mein Handy und Bobs Laptop nicht und können trotzdem (indirekt, über meinen Laptop) Daten austauschen. Bob sieht nicht, wann ich mit meinem Handy online bin. Falls ich nun doch mein Handy mit Bobs Laptop verbinden will, kann ich die Geräte miteinander bekannt machen und brauch bloss in allen zu teilenden Ordnern einen Haken zu setzen (und Bob dasselbe auf seinem Laptop). Da die Ordner und ihre Pfade, Namen und Einstellungen schon bekannt sind, braucht es keine weiteren Einstellungen als den Haken zu setzen. Selbstverständlich lässt sich das alles wieder rückgängig machen.

Die Syncthing-App für Android kann so eingestellt werden, dass sie entweder immer synchronisiert, nur per WLAN und / oder nur, wenn ein Ladegerät angeschlossen ist. Sobald sie läuft, überprüft sie alle überwachten Ordner auf Änderungen und guckt, ob ein bekanntes Gerät gerade erreichbar ist. Ist dies der Fall, werden die eingestellten Ordner auf die eingestellten Geräte synchronisiert. Der Akkuverbrauch ist dabei überraschend niedrig und auf meinem Handy gar nicht spürbar (Einstellung WLAN-only).

Die Web-GUI ist sehr modern, responsive und existiert auch auf dem Handy (in vollem Leistungsumfang!).

Jeder synchronisierte Ordner muss einen «Folder Marker» enthalten. Das ist eine leere versteckte Datei mit dem Namen .stfolder. Will man die Synchronisation eines Ordners pausieren und hat gerade keinen Bock, in die Config zu gehen, so löscht man einfach den Marker und Syncthing hört auf, den Ordner zu verwalten. Sobald man die Datei wieder erstellt, wird die Synchronisation fortgesetzt. ➔



Junior Software Engineer

Hast du einen ETH-Abschluss in Informatik und Java-/Java-EE-Kenntnisse? Interessierst du dich für Applikationsentwicklung und Integration?

Praktikum/Werkstudium

Machst du deinen Master in Informatik an der ETH? Willst du deine Kompetenzen bereits heute praktisch einsetzen?

Bewirb dich online!
HR-Team Schweiz
+41 44 272 61 11
jobs@adnovum.ch
adnovum.ch/jobs

kununu TOP COMPANY

Zürich · Bern · Lausanne · Budapest · Lissabon · Ho-Chi-Minh-Stadt · Singapur

Mindestens unter Linux unterstützt Syncthing via Plugin (welches im Paketmanager bequem installiert werden kann) inotify. Dies bedeutet, dass das Programm nicht pollen / aktualisieren muss, um Änderungen im Dateisystem zu erfahren: Das Betriebssystem benachrichtigt Syncthing, sobald etwas verändert wurde. Damit bleibt Syncthing über Stunden fast komplett inaktiv, wenn nichts läuft, und verbraucht so gut wie keine Ressourcen, wenn gerade nichts zu synchronisieren ist.

Geteilte Ordner verfügen über diverse Einstellungen:

- Scanintervall: Falls kein inotify aktiviert ist, scannt Syncthing den Ordner so häufig auf Veränderungen
- Automatisches Limit zum Stoppen der Synchronisation, falls nur noch n% Festplattenspeicher verfügbar sind
- Festlegen der Dateiübertragungsreihenfolge (Zufall / kleinestes / grösstes / neustes / ältestes / alphabetisch)
- Festlegen, ob Dateiberechtigungen zu ignorieren sind (z. B. wenn ein Ordner auf einer FAT-Partition liegt)
- Ordnertyp: Es ist möglich, zweiseitig zu synchronisieren, oder aber auf einem Host H einen Schreibschutz einzuschalten: Dabei werden Änderungen nur von H an andere übertragen, aber andere können den Ordner auf H nicht ändern. Natürlich macht es keinen Sinn, den Schreibschutz für denselben Ordner auf mehreren Geräten zu aktivieren (ansonsten kann man sich auf ein hübsches rotes «nicht synchronisiert» gefasst machen). Versuchen andere Geräte, wider Schreibschutz Änderungen reinzubringen, genügt ein Klick auf «Änderungen überschreiben».
- Ignoriermuster: Ähnlich wie eine .gitignore-Datei legt diese Einstellung fest, welche Dateien nicht synchronisiert werden sollen.
- Dateiversionierung: Dies erlaubt ein automatisches Backup der verwalteten Dateien. An diesem Punkt muss darauf hingewiesen werden, dass die Entwickler Syncthing explizit als «keine Backuplösung» anpreisen und nichts garantieren. Die Dateiversionierung funktioniert gemäss meiner Erfahrung jedoch sehr zuverlässig. Möglichkeiten sind:
  - keine Versionierung
  - Papierkorb-Versionierung: Sobald eine Datei gelöscht wird, wird sie N Tage lang behalten (N kann auch 0 für unendlich sein)
  - Einfache Dateiversionierung: Es werden die N neusten Versionen der Datei behalten
  - Stufenweise Dateiversionierung: In der ersten Stunde wird alle 30 Sekunden eine Version behalten, am ersten Tag eine jede Stunde, in den ersten 30 Tagen eine jeden Tag. Danach wird bis zum angegebenen Höchstalter eine Version pro Woche behalten. Dabei können Höchstalter und Pfad für die Versionierung frei gewählt werden
  - Externe Versionierung: Das ist richtig toll: Selber ein Skript schreiben und dessen Pfad hier angeben. Dann werden als erster Parameter der Ordner und als zweiter Parameter der relative Pfad im Ordner übergeben und man kann damit machen, was immer man möchte.

Wird das Dateisystem gerade nicht verändert, hat Syncthing einen sehr kleinen Fussabdruck und verursacht so gut wie keine CPU- und RAM-Auslastung (auf meinem Laptop 0 % CPU und 52 MB RAM bei fast 100 GB Datenvolumen und knapp 10'000 überwachten Files).

Hat man ein Problem, werden Syncthing-Issues auf Github meist sehr schnell und kompetent (oft sogar von den Entwicklern höchstpersönlich) beantwortet.

➔

Hier ein Überblick über die Infrastruktur, auf der ich Syncthing betreibe:

- Mein Handy mit Android
- Mein Arch-Linux-Laptop
- Mein Raspberry Pi in Zürich, mit externer Festplatte als Datenspeicher und Beamer / Stereoanlage als preisgünstiges Heimkino und stete Konkurrenz zu Alines Setup (siehe ihren Artikel in dieser VISIONEN-Ausgabe und ja, meine Anlage war zuerst ;-)).
- Ein Raspberry Pi, das ich meinen Eltern ins Haus gestellt habe. Es hat auch eine USB-Festplatte und hängt per HDMI am Fernseher meiner Eltern. Sie nutzen Kodi, um ihre Ferienfotos anzuschauen.
- Der Windows-PC meiner Eltern
- Der Laptop meiner besten Freundin Aline. Mit ihr hab ich bei einem Event mal eine grössere Menge an Filmmaterial geschossen, welches ich mit ihr teilen möchte.

Hier ein Teil der Ordner, die ich per Syncthing in der Schweiz herumverteile:

- Auf dem Handy geschossene Fotos und Videos (Ordner DCIM) gehen an meinen Laptop und an mein Raspberry Pi in Zürich. Sobald auf meinem Handy der Speicher ausgeht, verschiebe ich die Bilder auf meinem Laptop in einen anderen Ordner, der nicht mit dem Handy synchronisiert ist und schon hat mein Handy wieder Platz, ohne dass ich auf dem winzigen Display herausfinden muss, welches Bild denn nun das bessere ist und welches gelöscht werden soll. Wenn ich mal aus Versehen ein Bild zuviel lösche, ist das kein Problem: Mein Raspi hat eine Papierkorb-Versionierung, welche die aus diesem Ordner gelöschten Dateien 100 Tage behält.
- Filme, die ich auf meinem Heimkino ansehen möchte: DVD in den Laptop schieben und rippen. Schon springt Syncthing an und repliziert die Filme auf mein Raspi in Zürich und auf das im Haus meiner Eltern. Das ist praktisch, wenn ich einen Film für Zürich runterlade, aber dann unerwartet zu meinen Eltern reise: Der Film kommt schon vor mir dort an, ohne dass ich einen Finger rühren muss.
- Die Ordner für Fotos, Dokumente und Musik auf meinem Laptop enthalten Dateien, die ich nicht verlieren will. Deswegen werden sie von Syncthing automatisch auf mein Raspi geladen, welches sie versioniert, für den Fall, dass ich beim Tippen eines Befehls mit rm ausrutsche. Falls mir jemand mein Raspi hackt und alles löscht, muss ich mir keine Sorgen machen: Die Ordner sind auf meinem Laptop schreibgeschützt und können von keinem anderen Gerät verändert werden. Das Datenvolumen ist gewaltig, doch dank inotify kein Problem.
- Der Foto-Ordner meiner Eltern: Sobald Papa seine Ferienbilder in seinen PC speist, landen sie schon auf dem Raspberry Pi bei ihm und sind vor einfachem Festplattenausfall geschützt. Auch kann er einfach den Fernseher einschalten und die Bilder sind schon magisch auf Kodi gelandet. Falls WannaCry seinen Computer befällt, keine Sorge! Syncthing behält eine Kopie aller gelöschten Dateien für 100 Tage.
- Das Archiv meiner Eltern: Sobald die externe Archiv-Festplatte angeschlossen wird, springt Syncthing an und überträgt alle Änderungen aufs Raspi mit Kodi. Auch hier eine Doppelnutzung als Medienwiedergabe sowie «short-term-Backup» (siehe den Abschnitt mit den Limitierungen) mit Schreibschutz und Versionierung. Dabei handelt es sich um ein Datenvolumen von über 600 GB in über 200'000 Dateien, welches per verschlüsseltem Block Exchange Protocol an eine USB-Festplatte an einem Raspberry Pi übertragen wird! Kein Problem mit Syncthing: es berechnet die Prüfsummen und überträgt nur neue Daten. Dank Syncthings Multithreading geht das auf dem PC zügig und das Raspi ist meistens idle, weil es inotify hat und darauf ausser von Syncthing nichts verändert wird.
- Die Filmaufnahmen, die ich mit Aline gemacht habe, nehmen viel Platz ein, weil ich sie für Slow-Mo mit hoher Bildrate aufgezeichnet habe. Deswegen will ich sie schnellstmöglich von meinem Handy entfernt haben. Wie oben erwähnt ist mein DCIM Ordner mit meinem Laptop synchronisiert. Nun kann ich einfach die Aufnahmen auf dem Laptop vom Handy-DCIM-Ordner in den «Aufnahmen»-Shared-Ordner verschieben. Die Operation dauert einen Bruchteil einer Sekunde, da es dieselbe Partition ist. Sofort löscht Syncthing die Aufnahmen auf meinem Handy und überträgt sie auf den Aufnahmen-Ordner auf meinem Raspberry. Nun kann ich den Laptop ausschalten. Sobald Aline das nächste Mal mit ihrem Laptop online geht, erhält sie die Aufnahmen im Hintergrund. Falls mein Laptop dann zufällig auch gerade läuft, erhält sie die Daten gleichzeitig von meinem Laptop und meinem Raspberry, wodurch sie vom kombinierten Upstream profitieren kann.

All dies läuft sehr zuverlässig im Hintergrund und benötigt, einmal aufgesetzt, ein Minimum an Wartungsaufwand. Syncthing bewältigt grosse Datenvolumen selbst auf einem Raspberry Pi mühelos.

Download	1 B/s (1.55 GiB)
Upload	1 B/s (282 MiB)
Lokaler Status (Gesamt)	228078 Objekte, ~629 GiB
RAM Auslastung	147 MiB
Prozessorauslastung	0%
Zuhörer	2/2
Gerätesuche	5/8
Betriebszeit	1d 19h 41m
Version	v0.14.29, Linux (ARM)

Limitierungen

Ein paar Dinge müssen gesagt werden. Erstens ist die Sprache Go, in der Syncthing geschrieben ist, leider unter neueren Android-Versionen nicht fähig, die externe SD-Karte zu beschreiben. Syncthing macht schon seit einer Weile erfolglos Druck auf die Go-Entwickler [2]. Zweitens ist bei Syncthing nie ganz klar, wie lange etwas dauern wird. Manchmal dauert es eine Sekunde, bis sich zwei Geräte sehen, manchmal zwei Minuten. Drittens ist Syncthings Performance auf einem Raspberry Pi dadurch beschränkt, dass es keine Hardware-Unterstützung für die verwendete Verschlüsselung gibt. Der Durchsatz liegt bei 3 MB/s in beide Richtungen. Dies ist für mich jedoch kein Problem, da ich wie im Use Case erwähnt zwei Raspis habe, deren Performance sich addiert. Dadurch erhalte ich 6 MB/s im Downstream. Auch geschieht alles diskret im Hintergrund, während ich arbeite, weswegen mir egal ist, wie lange die Vorgänge dauern. Zuletzt muss erwähnt werden, dass Syncthing wie erwähnt gemäss den Entwicklern keine Backuplösung ist. Aline hat bei einer Misskonfiguration mal signifikanten Datenverlust erlitten. Syncthing ersetzt kein Backup. Auch ich mache periodisch Backups auf externen HDDs, was ich auch dringend empfehle. Syncthing fungiert bei mir nur als «short-term backup» für die allerneusten Änderungen.

Hintergrund und Technologie

Syncthing wurde und wird vom Schweden Jakob Borg entwickelt, ganz offensichtlich unter dem Motto «Do one thing, do it right!». Es ist in Go geschrieben, läuft unter Linux, Mac, Windows, Android, BSD und Solaris und unterstützt so ziemlich jede Architektur, auf der Go läuft, darunter eben auch das sagenhafte Raspberry Pi. Syncthings Lizenz ist die MPL 2.0 (Mozilla Public License). Die Software wird stetig (für besserwisserische Super-Nerds: ja, eigentlich diskret, aber das hier ist Deutsch und nicht Mathe)

weiterentwickelt und bringt in angenehmen Abständen fleissig Updates raus. Früher mal war es so, dass Versionen teilweise nicht miteinander kompatibel waren, doch dies geschieht jetzt, wo Syncthing reifer ist, nicht mehr.

Egal ob lokales Netz oder Internet – Syncthing rules them all! Es funktioniert mit seinem Discovery-Protokoll komplett offline, zum Beispiel wenn du im Zug bist und dein Handy und deinen Laptop per Hotspot miteinander verbindest. Sobald der Zug aus dem Funkloch fährt, syncen dann beide Geräte über Syncthings Discovery Server (oder deine eigenen, wenn du möchtest) mit deinem Host am anderen Ende der Welt. Kommt keine direkte Verbindung zustande, so nutzt Syncthing Relays, mit denen es über das Hindernis (z. B. NAT) hinwegsendet. All dies selbstredend verschlüsselt, und zwar mittels TLS. Beide IP-Versionen 4 und 6 werden unterstützt.

Syncthing hat für mich in den letzten zwei Jahren eine neue Ära der Dateiverwaltung ausgelöst.

Anbindung an Syncthing ist über diverse Methoden möglich: Die hübsche, moderne Web-UI ermöglicht bequeme Konfiguration per Browser. Diese kann auch angewiesen werden, anstatt nur auf localhost (127.0.0.1) auf beliebige Adressen (0.0.0.0) zu hören. Nun kann ein headless Server problemlos vom LAN aus konfiguriert werden, als würde er auf der eigenen Kiste laufen. Der Zugang kann selbstverständlich passwortgeschützt werden. Das klappt sogar auf dem Handy! Sehr praktisch für mich, denn ich hasse Touchscreens, weswegen ich grössere Änderungen auf der Syncthing-Konfig meines Handys gern am Computer durchführe. Für API-Fans gibt es eine REST- und eine Events-API, welche gut dokumentiert und mittels API-Key gesichert ist. Docker-Images sind ebenfalls verfügbar und Puppet, Ansible und Co. werden unterstützt. Zu guter Letzt ist das Config-File menschenlesbar, was bedeutet, dass man Syncthing stoppen, die Datei bearbeiten und Syncthing wieder starten kann (sehr praktisch besonders für Linux-Nerds, die gern in der Konsole rumhacken).

Fazit

Syncthing ist genial. Es hat für mich in den letzten zwei Jahren eine ganz neue Ära der Dateiverwaltung ausgelöst. Es ist so anpassungsfähig, dass es so gut wie jeden Use Case abdecken kann. Seine Zuverlässigkeit und Stabilität, besonders in letzter Zeit, wo das Programm reifer geworden ist, erstaunt mich immer wieder und lässt Hoffnung in die oft so kaputte Softwarewelt aufkommen. Als diskreter Helfer im Hintergrund greift mir Syncthing bei der Dateiverwaltung im grossen Stil unter die Arme und läuft auf jeder noch so crappy Hardware, die ich ihm antue. Die Entwickler haben es geschafft, die Aufgabe des Programms in vollem Umfang zu verstehen und haben eine Software geschrieben, die sich in ihrem Job in ihrem Element fühlt. Syncthing wird mich wohl mein Leben lang begleiten.

</hype>

Ein Tag im Leben eines Chemiestudenten

ADELAIDA REDZEPOVIÇ – HAT DAS FALSCHE STUDIENFACH GEWÄHLT :P

Als Informatikerin oder Informatiker verirrt man sich selten mal auf den Campus Hönggerberg, und noch seltener in die Höhle des ~~Löwen~~ Chemikers, unser heissgeliebtes HCI-Gebäude: da, wo Chemikerträume in den Laboren wahr werden. Ein Gebäude, in dem es von Labormantel-tragenden, verwirrten angehenden Wissenschaftlern nur so wuselt. Wenn nicht gerade in einem Labor sämtliche Chemikalien, sei es absichtlich oder aus Versehen, in die Luft gesprengt werden, geht es bei uns eigentlich ganz friedlich zu und her.

Und jetzt mal Vorurteile beiseite: Wir laufen tatsächlich mit mehr oder minder normaler Alltagskleidung rum, wenn wir nicht gerade den Praktika nachgehen und es wird (leider) nicht grossartig was zur Explosion gebracht. Die Zeit neben den Vorlesungen und Übungen, die ein Durchschnittsinformatiker wohl am Computer verbringt, verbringen wir in unseren heissgeliebten Laboren, da, wo wir die Möglichkeit haben, Glaswaren kaputt zu machen, die gefühlt mehr kosten als unsere monatliche Miete.

Während ihr an euren Algorithmen rumtüftelt, schreiben wir Berichte über Versuche, die in die Hose gegangen sind (nein, es landen nicht wortwörtlich Chemikalien in der Hose – so ungeschickt sind wir nun auch wieder nicht) und werten etliche Daten aus, wobei wir uns fast schon wie richtige Informatiker fühlen, wenn wir mit R unser eigenes Skript zur Datenauswertung verfassen und benutzen.

Unsere Freizeit verbringen wir auch nicht damit, neue Synthesewege zu finden oder das Periodensystem auswendig zu lernen, wobei ich hierbei nur über den Durchschnittschemiker schreibe – die ETH hat, wie ihr selber bestimmt tagtäglich erlebt, viel Skurriles zu bieten, also bestimmt auch den einen oder anderen Chemiker, der das Periodensystem besser kennt als seine eigene Hosentasche.

Wenn wir nicht gerade Chemikalien in die Luft sprengen, geht es eigentlich ganz friedlich zu und her.

Unsere ausserordentlichen Fähigkeiten sind überall gefragt - wir wissen, wie man feiert und sind mit unserem selbergemachten Stickstoff-Eis gern an Events gesehen, denn was gibt es cooleres (pun intended) als einen Chemiker, der mit scheinbar viel Tamtam und Rauch ein feines Eis herbeizaubert?

So, und nun fertig mit Artikel schreiben, ich muss zurück ins Labor, bevor es mir die ganze Kapelle in die Luft jagt. Und kommt ruhig Hallo sagen, falls ihr euch mal nach Science City verirrt!

Never Heard of It #18

BALZ GUENAT – SHOULD REALLY TRY TO FIND SOME NORMAL MUSIC FOR A CHANGE ...

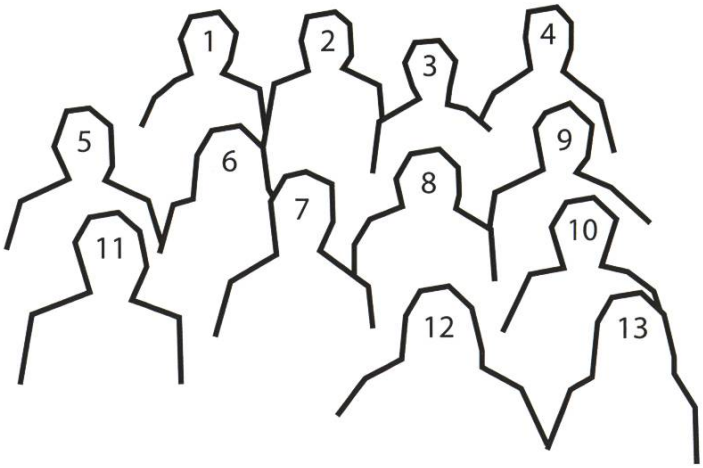
At the moment of writing, the E3 gaming conference has just passed, and while some things should've been big news, like the announcement of the 4K-capable Xbox, the only memorable thing for me was the "conference" of Devolver Digital, the publisher probably best known for the Hotline Miami games. Their parody of an E3 press conference was not only as insane as those games, but somehow also bloodier. A fact that seems impossible if you know the games. Oh yeah, and of their 15-minute show, they also used about 90 seconds in total to show off two of their upcoming games. One of those games was Ruiner, and the short segment about it had some really cool music. That is what I want to talk about today.

Sidewalks and Skeletons – White Light

As every so often, this is some weird stuff. To start off, the name White Light is pretty misleading. The album has a very dark and somewhat creepy tone to it. If I had to guess how to call this genre of music, I'd have said Horrorwave or something. Looking it up reveals that apparently the genre is called Witch House. I'd describe it as droning music with synth and organ sounds accompanied by simplistic drum machine beats. The influences can be found all over the place: Synthwave, Metal, Techno, Trap, Future Bass and Vaporwave. It's a great big awesome mess and feels like a neon-lit fever dream! Ah, by the way, the track that played to Ruiner was Sleep Paralysis, in case you were wondering.

Year: 2015
Length: 44:14
Spotify:
tinyurl.com/NeverHeardSpotify
Google Play Music:
tinyurl.com/NeverHeardGoogle

Visionäre



Blockieren gerne Treppen: Noah Delius[1], Dominic Sonderegger[2], Silvia Siegrist[3], Pascal Wacker[4], Aline Abler[5], Sarah Kampf[6], Andreas Brombach[7], Balz Guenat[8], Mickey Vänskä[9], Zeno Koller[10], Robin Bisping[11], Julian Croci[12] und Lena Csomor[13].

Impressum

VISIONEN

Magazin des Vereins der Informatik Studierenden an der ETH Zürich (VIS)

Ausgabe September 2017

Periodizität 6x jährlich
Auflage 1800

Chefredaktion
Zeno Koller
chefredaktor@vis.ethz.ch

Cover
Layout-Team

Layout
Aline Abler
Nicolas Winkler
Robin Bisping
layout@vis.ethz.ch

Inserate
Balz Guenat
inserate@vis.ethz.ch

Anschrift Redaktion & Verlag
Verein Informatik Studierender (VIS)
CAB E31
Universitätsstr. 6
ETH Zentrum
CH–8092 Zürich

Inserate (4-farbig)

½ Seite	CHF 1000.–
1/1 Seite	CHF 1800.–
2/1 Doppelseite	CHF 4000.–
1/1 Seite, Umschlagsseite (U2)	CHF 3000.–
½ Seite, Rückumschlag (U4)	CHF 3000.–

Andere Formate auf Anfrage.

Lektorat
Felice Serena
Dominic Sonderegger
Nils Leuzinger
Noah Delius
Quentin Hibon
lektorat@vis.ethz.ch

Redaktion
Julian Croci
Stefan Dietiker
Manuel Braunschweiler
Zeno Koller
Ganesh Ramanathan
Lukas Widmer
Andreas Brombach
Sarah Kamp
Pascal Wiesmann
Michael Grossniklaus
redaktion@vis.ethz.ch
und freie Mitarbeiterinnen und Mitarbeiter

Druck
Sprüngli Druck AG
5612 Villmergen
http://www.spruenglidruck.ch/

Copyright
Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des VIS in irgendeiner Form reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Offizielle Mitteilungen des VIS oder des Departements für Informatik sind als solche gekennzeichnet.

© Copyright 1989–2016 VIS. Alle Rechte vorbehalten.

Die Visionen werden klimaneutral gedruckt.

Mix
Papier aus verantwortungsvollen Quellen
FSC® C007061

Swiss Climate Klimaneutral gedruckt

VSETH VERBAND DER STUDIERENDEN AN DER ETH

Der VIS ist Teil des Verbandes der Studierenden an der ETH (VSETH).

Die Welt gemäss Beni Koller

Zunderspiele

MICHAEL GROSSNIKLAUS – EIN MONUMENT

Es ist eine dieser tropisch heissen Sommernächte. In der Hoffnung nach ein bisschen Abkühlung hatte Beni alle Fenster seiner Wohnung sperrangelweit geöffnet. Doch der erlösende Durchzug bleibt aus, da die schwüle Luft an diesem Abend gänzlich stillsteht. Beni liegt splitternackt auf dem Rücken im Bett und kriegt kein Auge zu. Er spürt, wie sich die Schweissperlen auf seiner Brust in der Vertiefung über dem Sternum sammeln und von dort als kleines Rinnsal zum Bauchnabel fliessen. Doch es ist nicht nur die drückende Hitze, die Beni in den letzten Nächten den Schlaf geraubt hat: Vor ein paar Tagen hat Marie mit ihm Schluss gemacht.

Beni versucht zunächst, sich mit einem Buch abzulenken. Normalerweise klappt das gut, sodass er nach nur wenigen Seiten einschläft. Obwohl ihn das Buch «Mr. Penumbra's 24-Hour Bookstore», das er gerade liest, sehr fasziniert, kann er sich nicht auf die Handlung konzentrieren, da seine Gedanken immer wieder zur Trennung von Marie abschweifen. Beni beschliesst daher, es mit leichterer Kost zu versuchen, und greift zum iPad, das auf dem Nachttischchen neben dem Bett liegt. Er beginnt, seinen Feed auf Facebook zu durchstöbern, bringt sich mit Schlagzeilen der Washington Post und der New York Times auf den neusten Stand über die Kapriolen des U.S.-Präsidenten, staunt über 12 Alltagsgegenstände, die er anscheinend bisher immer falsch benutzt hat, und schaut sich ein paar Clips aus Late Night Talkshows an.

In einem dieser Clips reisst der Host der Talkshow im Eröffnungsmonolog derbe Scherze über Tinder-Dates. Aus einer merkwürdigen Mischung von Neugier, Experimentierfreude und Einsamkeit heraus, lädt sich Beni die Tinder-App auf sein iPhone und beginnt diese zu konfigurieren. Ein paar clevere und selbstironische Zeilen, die ihn beschreiben, hat er schnell geschrieben. Danach muss er auswählen, welches Bild aus seinem Facebook-Profil bei Tinder erscheinen soll. Die meisten seiner Profilbilder bei Facebook sind im Spass entstanden und daher eher nicht für eine Datingplattform geeignet. Bei denen, die infrage kommen, kann sich Beni nicht entscheiden, welches das Beste ist. Zum Glück erinnert er sich daran, dass er kürzlich von einer Webseite gelesen hat, die behauptet, mithilfe von Machine Learning die Attraktivität von Personen in Profilbildern bestimmen zu können. Seit jeher technologiegläubig vertraut Beni dem Urteil dieser Webseite und wählt das Bild, mit dem er am besten abschneidet, als Tinder-Profilbild.

Nachdem das Tinder-Profil endlich komplett ist, kann Beni mit dem Swipen beginnen. Bei den meisten Profilen ist Beni schlicht verblüfft, wie schlecht sich die jeweiligen Personen präsentieren, sodass er mit dem Finger meistens nach links wischt. Es dauert mehr als 20 Profile bis Beni jemanden sieht, der sein Interesse weckt. Das Profilbild zeigt eine junge Frau, die auf einem Mountainbike einen steilen Berg hinunterrast. Auf dem zweiten Bild erkennt Beni, dass die Frau auch ohne Integralhelm gut aussieht. Als er jedoch liest, dass sie sich als Katzenmensch outet, wischt er schon fast instinktiv wieder nach links. Einen Moment lang wundert sich Beni über seine eigene Oberflächlichkeit, findet sich dann aber mit der Einsicht ab, dass das wohl zu Tinder gehört und seine Gegenüber das auch nicht anders handhaben.

Beni muss sich erneut durch etliche uninteressante Profile swipen, bis wieder eines kommt, das ihm ins Auge sticht. Das Profilbild zeigt eine blonde Frau namens Christa mit einem breiten Lachen, das Beni sofort sympathisch ist. Ohne weiter zu zögern, wischt Beni dieses Mal nach rechts und ist gespannt, was passiert. Da unmittelbar gar nichts passiert und ihm bereits schon wieder ein völlig uninteressantes Profilbild angezeigt wird, legt Beni das iPhone zur Seite und versucht nochmals zu schlafen.

Als Beni am nächsten Morgen die ersten Sonnenstrahlen nach einer unruhigen Nacht wecken, fühlt er sich noch überhaupt nicht erholt und völlig kaputt. Sein erster Gedanke ist deshalb nicht, aufs iPhone zu sehen und zu prüfen, ob sich bei Tinder etwas getan hat, sondern so schnell wie möglich Koffein in seinen Körper zu bekommen. Erst nachdem er einen Latte Macchiato mit drei Shots Espresso getrunken hat, fällt ihm wieder ein, wie er die schlaflose Phase der letzten Nacht verbracht hat. Neugierig schlurft er zurück ins Schlafzimmer und dreht sein iPhone um.

Auf dem Display werden zwei Nachrichten angezeigt. Die erste informiert ihn, dass über zehn Leute an ihm interessiert seien und dass er weiterwischen soll, um zu erfahren, ob die Anziehung auf Gegenseitigkeit beruhe. Nach allem, was Beni letzte Nacht so auf Tinder gesehen hatte, schätzt er die Chance, dass dem so ist, als eher gering ein. Viel interessanter für ihn ist die zweite Meldung, die ihn über einen neuen Match informiert. Da Beni nur einmal nach rechts gewischt hatte, muss es sich dabei um Christa handeln. Als er die Tinder-App öffnet, um diese Schlussfolgerung zu bestätigen, ermutigt ihn diese, Christa eine Nachricht zu schreiben. Auf einen Schlag wird Beni klar, dass diese Situation völlig neu für ihn ist. Noch nie zuvor hatte er sich mit einer Frau unterhalten, ohne sehen zu können, wie seine Annäherungsversuche bei ihr ankamen. Beni starrt minutenlang auf den leeren Chatverlauf und die darunterliegende Tastatur, bevor er Christa ein bescheidenes «Hallo! Wie geht's?» schickt.

Fortsetzung folgt …

AZB
PP/Journal
CH – 8092 Zürich

Falls unzustellbar, bitte zurück an:
Verein der Informatik Studierenden
CAB E31
Universitätsstr. 6
ETH Zentrum
CH-8092 Zürich