

Was macht ein Legal Engineer?

Ein Interview mit Sebastian Nagl



Sebastian Nagl arbeitet als Legal Engineer für die internationale Wirtschaftskanzlei Hogan Lovells und lehrt als Dozent für IT-Recht, Rechtsinformatik und Legal Tech am Rechtsinformatikzentrum (RIZ) der Ludwig-Maximilians-Universität München. Er hat sein erstes Staatsexamen absolviert und befindet sich aktuell im Referendariat. Er hat unter anderem die Munich Legal Tech Student Association e.V. (MLTech), eine studentische Organisation für Law & Tech in München, und den Open Legal Tech e.V., einen Verein zur Verbreitung von Open Source Legal-Tech-Lösungen, mit aufgebaut.

CTRL: Die wichtigste Frage vorab: Was genau ist ein Legal Engineer?

Es gibt noch keine klare Definition davon, was ein Legal Engineer ist. Für mich und die Art und Weise wie meine Kanzlei das Berufsbild versteht, ist ein Legal Engineer jemand, der versucht juristische Probleme und Sachverhalte durch technische Mittel zu optimieren bzw. zu vereinfachen. Konkret bedeutet das, dass man kanzleintern mit einem konkreten juristischen Problem auf uns zugeht und wir dann technisch darüber nachdenken und gemeinsam versuchen, einen Weg zu finden, dieses Problem zu automatisieren oder zumindest die juristische Lösung dieses Problems zu entschlacken.

CTRL: Was ist das Anforderungsprofil an einen Legal Engineer? Welche Skills muss ein Legal Engineer haben?

Ich beschreibe euch mal einen Legal Engineer, wie ich ihn persönlich gern in meinem Team hätte: Es gibt dieses Skillset, das ich selbst "heilige Dreifaltigkeit" eines modernen Juristen nenne: Teil eins, natürlich die juristische Kompetenz, die du brauchst, um die Sachverhalte zu durchdringen. Der Fokus läge für mich klar auf guter juristischer Methodenlehre.

Nummer zwei, die technischen Kenntnisse:

Solides Grundverständnis der Hardware und vielleicht das ein oder andere abgeschlossene Projekt in einer modernen Skriptsprache wie Python oder JavaScript. Drittens, ökonomisch zumindest ein Grundverständnis. Diese drei Komponenten sind in meinen Augen aber auch die idealen Voraussetzungen für quasi jeden modernen Juristen...

Für einen Legal Engineer liegt der Fokus verstärkt auf den technischen Kenntnissen. Die juristischen Kenntnisse treten bei dieser Arbeit ein wenig in den Hintergrund, aber du musst in der Lage sein, den Sachverhalt wirklich zu durchdringen und ein Verständnis für die juristische Schachtelprüfung zu haben. Optional, aber ideal ist ein wirtschaftliches Grundverständnis, um zum Beispiel eine Kosten-Nutzen-Analyse für potenzielle Projekte durchführen zu können. Darüber hinaus werden auch die für die Softwareentwicklung typischen Schlüsselkompetenzen benötigt. Du musst strukturiert arbeiten, damit du dir die Fülle an anfallenden Aufgaben aufteilen kannst. Optimal ist es, wenn du außerdem etwas von Projektmanagement verstehst. Auf jeden Fall muss ein Legal Engineer programmieren können. Für meine Begriffe ist er derjenige, der sich "die Hände schmutzig macht". Meine Hauptarbeit besteht darin, zu coden. Natürlich gehört auch die Besprechung der Herangehensweise an Projekte zu meiner Arbeit, aber ich würde sagen, dass ich zu 80 %, wenn nicht zu 90 % meiner Arbeitszeit am Programmcode sitze. Als Legal Engineer musst du technisch noch besser sein, als du juristisch bist.

CTRL: Wie muss man sich den Arbeitstag eines Legal Engineers vorstellen? Sie kommen in Ihr Büro, schalten Ihren Computer an und wie geht es dann weiter?

Morgens früh besprechen wir uns im Team. Dabei versuchen wir agile Methoden des Arbeitens zu integrieren. Ein Beispiel sind Stand-up Meetings: Jeder in der Runde erzählt kurz, woran er aktuell arbeitet und alle berichten, was gut bzw. nicht so gut gelaufen ist. Danach schauen wir uns die Aufgaben an, die wir zu erledigen haben, bewerten sie nach ihrer Komplexität und

erstellen Übersichten. So visualisieren wir, was wir schaffen wollen. Jeder bekommt seine Aufgabe und dann wird programmiert. Am Anfang dieser Programmierfähigkeit steht eine Problemanalyse. Wir fragen uns, wo genau das Problem liegt und versuchen, dieses Problem technisch oder juristisch zu strukturieren und zu lösen.

CTRL: Können Sie uns einen Einblick in Ihre Projektabläufe geben? Wie wird aus einer Idee ein fertiges Software-Produkt?

Projektabläufe sind von Unternehmen zu Unternehmen unterschiedlich ausgestaltet. In einem jungen Startup, das nur einen Legal Engineer beschäftigt, werden die Projektabläufe ganz anders aussehen als bei uns in der Großkanzlei.

In der Regel kommen die Projektideen von den Anwälten, die auf uns zukommen und sagen: „Wir haben regelmäßig folgendes Problem. Kann man da nicht was machen?“, „Ich hätte eine Idee!“ oder „Ein Mandant hat mich auf etwas hingewiesen. Lässt sich diese Problematik in einem Tool realisieren?“.

Im Anschluss setzen wir uns mit dem Ideengeber oder der Ideengeberin zusammen und versuchen mit ihm oder ihr das Problem zu ergründen, um festzustellen, wo der Schuh drückt. In einem zweiten Schritt überlegen wir, ob sich die Idee aus technischer Sicht umsetzen lässt. Hier pflegen wir eine sehr offene Kommunikation, denn es muss sich auch lohnen, für den jeweiligen Use Case eine Anwendung zu entwickeln.

Die Umsetzung einer Idee hängt auch von einer Kosten-Nutzen-Analyse ab, deren Bezugspunkte insbesondere die zu erwartende Arbeitszeit und der Umfang an benötigten Ressourcen sind. Im nächsten Schritt nehmen wir uns des Projektes an. In unserem Development-Team überlegen wir uns, wie wir die an uns gestellte Aufgabe herunterbrechen können. Bei der Aufgabenverteilung orientieren wir uns daran, wer worin besonders gut ist und wie der zeitliche Ablauf der Entwicklung aussieht.

In regelmäßigen Abständen geben wir Feedback nach außen und teilen dem jeweiligen Ideengeber den aktuellen Entwicklungsstand mit. Der Entwicklungsprozess vom Prototypen bis hin zum fertigen Softwareprodukt verläuft

zyklisch: stetig tauschen wir uns mit den Anwälten aus und entwickeln das Produkt weiter.

CTRL: Das heißt Ihre Arbeitsweise lebt vom ständigen Austausch?

Auf jeden Fall! Während der eigentlichen Entwicklungsphase versuchen wir zwar, externe „Störquellen“ auszuschalten, damit wir effektiv arbeiten können. Wir brauchen aber natürlich in regelmäßigen Abständen wieder Feedback!

CTRL: Dabei funktioniert die interdisziplinäre Zusammenarbeit gut? Sind Juristen offen genug für Neues?

Ja und Nein. Es gibt viele Juristen aus meinem Umfeld, die Legal Engineering höchst skeptisch beäugen, aber bei uns in der Kanzlei kommt unsere Arbeit sehr gut an. Wir können uns vor Anfragen kaum retten. Jeder hat eine Idee, einen Use Case, den er gelöst haben möchte. Das hat in sehr kurzer Zeit sehr stark an Fahrt aufgenommen.

Noch bis vor kurzem schien die Devise auf dem Rechtsmarkt zu sein: „Wir sollten uns irgendwie digitalisieren.“ Jeder hat sich Legal Tech auf die Fahne geschrieben, aber keiner wollte es so richtig anfassen. Jetzt, wo es zu funktionieren scheint, kommt es gut an und die Leute versuchen auf diesen Zug aufzuspringen. Die interdisziplinäre Arbeit in der Kanzlei funktioniert bei uns sehr gut. Sowohl Partner, als auch wir - also die Legal Engineers - sind auf einer Wellenlänge und auf Augenhöhe, wenn wir miteinander kommunizieren. Sie verstehen, dass wir Expertise haben, wir verstehen, dass sie Expertise haben. Gerade den Legal Engineers fällt es nicht schwer, die andere Seite zu verstehen, weil ihnen weder Technik noch Jura fremd ist.

Entscheidend für die gute Zusammenarbeit ist insbesondere der gegenseitige Respekt der Anwälte und Legal Engineers füreinander.

CTRL: Welche Art von Projekten steht bei Ihnen aktuell im Fokus?

Unsere Projekte sind recht vielseitig, wobei ich an dieser Stelle natürlich keine Anwendung im Einzelnen beschreiben darf. Um euch gleichwohl abstrakt eine Vorstellung der

Ausrichtung unserer Projekte zu geben: Das können ganz konkrete enge juristische Fallgestaltungen sein, also ein einziges juristisches Problem, das sich gut technisch abbilden lässt. Andererseits haben wir auch größere IT-Infrastrukturaufgaben, die im Raum stehen. In diesen Bereichen liegt viel Entwicklungspotential. Es gibt auch Fälle, in denen keine klassisch juristische Aufgabe gelöst werden muss, sondern ein Problem im Bereich der allgemeinen Kanzleiverwaltung. Wir versuchen die ganze Bandbreite an Problemfeldern abzudecken.

CTRL: Zeichnen sich für das Berufsbild Legal Engineering bestimmte Trends ab?

Ich glaube es gibt einen Trend dahingehend, dass alle größeren Kanzleien versuchen Legal Engineering, oder zumindest technische Expertise, zu integrieren.

Alle versuchen gerade, diese kleinen Teams aufzubauen, die neben dem klassischen hierarchischen Konstrukt in der Kanzlei - Associate, Senior Associate, Partner - stehen sollen. Keine Kanzlei traut sich die Potentiale des Legal Engineering „liegen zu lassen“. Ich glaube das ist ein Trend, der sich recht klar abzeichnet: Alle merken, dass man diesen Bereich nicht vernachlässigen sollte.

CTRL: Was ist die größte Herausforderung für einen Legal Engineer bzw. für Sie persönlich?

Die größte Herausforderung ist es, Wünsche und Erwartungen, die an mich als Legal Engineer gestellt werden, möglichst perfekt abzubilden.

Ich glaube, dass diese Herausforderung sich gleichermaßen in der rechtsberatenden Tätigkeit findet. Auch im Gespräch mit dem Mandanten geht es im ersten Schritt darum, das Problem herauszuarbeiten. Es ist die Kunst an dieser Schnittstelle, sei es zwischen Mandant und Anwalt oder zwischen Anwalt und Legal Engineer, möglichst gut zu kommunizieren, um möglichst genau das zu realisieren, was die jeweils andere Seite möchte. Ich würde sagen, das ist, auch wenn man sowohl juristische als auch technische Kenntnisse hat, immer noch eine Herausforderung.

Eine schöne und spannende Herausforderung, gleichwohl eine Herausforderung.

„Nochmal zur Klarstellung:
Ich identifiziere eine Person als einen Legal Engineer, wenn derjenige oder diejenige wirklich aktiv programmiert. Was ‚engineert‘ denn jemand, der nicht programmiert?“

CTRL: Sie haben bereits erwähnt, dass es einen Trend dahingehend gibt Legal Engineering-Teams in Kanzleien zu integrieren. Hat jede moderne Kanzlei heutzutage (schon) eigene Legal Engineers oder wer ist der „gewöhnliche“ Arbeitgeber für einen Legal Engineer?

Ich würde sagen, dass die beiden interessanten Bereiche für Legal Engineers Start-ups oder Großkanzleien sind. Ein Team aus Entwicklern oder Legal Engineers zu beschäftigen kostet Geld. Das machen kleinere Kanzleien in der Regel nicht.

Aus der Sicht eines Legal Engineers:
Entweder man stellt fest, dass man selbst ein einzigartiges Skill Set hat und gründet selbst ein Unternehmen. Dafür gibt es genug Use Cases. Jeden zweiten Tag, wenn du die Zeitung aufschlägst, siehst du irgendwas, wo du sagst „Okay, das kann ich mitnehmen“.
Oder du sagst, du willst an den großen Fällen arbeiten und mit finanzieller Rückendeckung die entsprechenden Anwendungen programmieren. Dann ist eine Großkanzlei wohl die bessere Alternative.

Ich gehe davon aus, dass tendenziell die meisten Großkanzleien Legal Engineers beschäftigen. Ob die meisten es schon so nennen, weiß ich nicht. Auf jeden Fall wird es in den meisten Großkanzleien ein Team von Informatikern geben, die Juristen dabei unterstützen ihre Arbeitsweise zu optimieren.

CTRL: Wie wird sich dieses Berufsfeld in der Zukunft entwickeln? Die Kritiker, die einen Legal Engineer aber als jemanden begreifen, der nicht selbst programmiert, gehen davon aus, dass es in der Zukunft genug Juristen geben wird, die hinreichend technisch versiert sind, sodass der Legal Engineer als Mittler zwischen dem Juristen und dem Programmierer wegfällt. Wie sehen Sie das? Werden wir später vielleicht gar alle als Legal Engineers arbeiten?

Nochmal zur Klarstellung: Ich identifiziere eine Person als einen Legal Engineer, wenn derjenige oder diejenige wirklich aktiv programmiert. Was „engineert“ denn jemand, der nicht programmiert?

Es gibt schließlich auch andere Berufsfelder, wie beispielsweise einen Legal Architect. Das ist jemand, der versucht die Sachen so zu konstruieren und aufzubereiten, dass der Programmierer oder Legal Engineer sie dann umsetzen kann. Das wäre ein Beispiel für eines der Berufsbilder im Bereich Legal Tech, für die man nicht zwingend aktiv programmieren muss. Aber nach meiner Vorstellung ist ein Legal Engineer jemand, der selbst programmiert. Ich glaube nicht, dass in absehbarer Zeit ein größerer Teil an Juristen in der Lage sein wird, vernünftig selbst zu programmieren und selbst Anwendungen zu entwickeln. So sehe ich das auch, wenn ich mir die Juristen bei uns in der Kanzlei ansehe. Brillante Köpfe, bei denen es schade wäre, wenn sie überhaupt anfangen zu programmieren, weil ihr Potenzial ganz klar darin liegt, juristisch zu arbeiten.

CTRL: In Schottland gibt es eine besondere berufliche Anerkennung für Legal Technologists. Bedarf es in Deutschland einer Anerkennung für eine besondere berufliche Qualifikation als Legal Engineer?

Ich finde es schwierig, dass Legal Engineer kein geschützter Begriff ist.

Es wäre schön, wenn man sich nur so nennen darf, wenn man ein bestimmtes Skill Set garantieren kann.

Aktuell ist das ein wahnsinnig schwammiger Begriff. Wir bewegen uns von „Ich habe mal an einem Hackathon teilgenommen.“, „Ich habe mal ein „hello world“- Programm geschrieben.“ bis hin zu „Ich arbeite eigentlich als Full Stack Developer.“ Alle diese Menschen können sich Legal Engineers nennen.

Für eine berufliche Qualifikation müsste man bestimmte Anforderungen festlegen. Ob ein Staatsexamen oder ein Studienabschluss in Informatik erforderlich sein sollte, kann ich nicht festlegen, aber ich würde mir wünschen, dass es auf irgendeine Art und Weise standardisiert wird. Es wäre schön, wenn es etwas gäbe, woran man die hinreichende Qualifikation als Legal Engineer objektiv festmachen könnte, also zumindest bestimmte Coding-Kurse, die man dann belegt haben muss, oder eine offene Prüfung, die man ablegen kann.

„Accredited Legal Technologist“

Die Law Society of Scotland verleiht den Titel des „Accredited Legal Technologist“ an schottische Rechtsanwälte, Rechtsanwaltsfachangestellte und andere, in der Entwicklung von Rechtstechnologie, tätige Fachleute, die über hervorragende Fachkenntnisse auf dem Gebiet der Rechtstechnologie verfügen. Nach eigener Aussage zielt die Law Society of Scotland darauf ab, die Entwicklung neuer Berufsbilder im Bereich Legal Tech zu fördern. Kriterien für eine Akkreditierung sind etwa die Leitung und strategische Entwicklung von Technologieprojekten in der Rechtspraxis oder eine besondere Expertise in bestimmten Feldern der Rechtstechnologie.

CTRL: Damit sind wir wieder bei der Frage nach den Zukunftsaussichten des Berufsfeldes. Muss man nicht die Frage in den Raum stellen, wie sich dieses Berufsfeld überhaupt vernünftig entwickeln kann, wenn es keine besondere Anerkennung der beruflichen Qualifikation dafür gibt?

Ja, das ist richtig. Auf der anderen Seite muss bedacht werden, dass wir uns in einem IT-Kontext bewegen, in dem die besten Leute, die ich kenne, Autodidakten sind. Das heißt, diese Leute können kein Studium im Bereich Informatik vorweisen, sind aber großartige Programmierer. Dementsprechend kann bzw. sollte es wahrscheinlich nicht an Studienabschlüssen gemessen werden. Es muss einen Erwartungshorizont in Bezug auf ein bestimmtes Skill Set geben, das sich am Markt durchsetzt, sodass die Leute sich erst Legal Engineer nennen können, wenn sie zum Beispiel die Zwischenprüfung und einen höheren Coding-Kurs geschafft haben. In jedem Fall begrüße ich die Möglichkeit einer Anerkennung als Legal Technologist in Schottland. Großartig, dass es dort so etwas gibt.

CTRL: Wo sehen Sie das größte Potenzial Abläufe durch Legal Engineering zu verbessern?

Als Legal Engineer für eine Großkanzlei muss ich mir anschauen, womit die Juristen bei uns ihre Zeit verbringen bzw. womit sie zu viel Zeit verbringen.

Das Ziel ist es, dass die Anwältinnen und

Anwälte möglichst viel juristisch arbeiten, denn das kann ihnen keiner abnehmen. Darin sind sie brillant. Sie sollten möglichst wenig andere Sachen machen. Das heißt, in meiner Vorstellung versuche ich die Arbeitszeit dahingehend zu optimieren, dass die Anwältin oder der Anwalt sich auf die Arbeit für den Mandanten konzentrieren kann.

Auch der freie Markt bietet sehr viel Potential für Legal Engineering, insbesondere in den Fällen, in denen sich Recht einfach automatisieren lässt. Ein Beispiel ist das bekannt gewordene "Datenleck" bei Scalable Capital. Einen Tag nach dem "Datenleck" waren schon Seiten online, die für die von dem Datenschutzvorfall bei Scalable Capital betroffenen Kunden Legal-Tech-Angebote formuliert haben. Das ist ein Aspekt, der nicht unterschätzt werden sollte: Wer aktiv Zeitung liest, stellt schnell fest, dass die Welt voller Möglichkeiten ist.

Der „Datenschutzvorfall“ bei Scalable Capital

Scalable Capital ist ein sogenannter Robo-Advisor. Robo-Advisor sind Online-Vermögensverwalter, die automatisiert Gelder von Anlegern investieren. Wie die Süddeutsche Zeitung berichtete, informierte Scalable Capital seine Kunden am 19. Oktober 2020 über einen „Datenschutzvorfall“: Es sei zwar kein Geld abgeflossen, aber „unrechtmäßig auf einen Teilbestand von Dokumenten zugegriffen“ worden, heißt es in der Nachricht an die Kunden. Betroffen waren personenbezogene Daten von insgesamt 20.000 Kunden: Jemand habe sich Zugriff auf Kontaktdaten, Ausweiskopien, Steuernummern, Wertpapierabrechnungen und Kontonummern verschafft. Bereits einen Tag, nachdem der Vorfall bekannt geworden war, boten die Legal Tech-Anbieter „Kleinfee“ und die „Europäische Gesellschaft für Datenschutz“ betroffenen Scalable-Kunden eine Geltendmachung ihrer ggf. bestehenden Auskunfts-, Unterlassungs- und Schadensersatzansprüche nach der DSGVO an.

CTRL: Haben Sie Tipps für unsere LeserInnen, die sich dafür interessieren Legal Engineers zu werden? Sie hatten bereits angesprochen, dass es eine ganze Reihe von hoch qualifizierten Legal Engineers gibt, die Autodiktaten sind.

Der allerwichtigste erste Schritt ist es, sich mit anderen Leuten zusammenzuschließen, die diesen Bereich spannend finden und die nicht

sagen „Jura bedeutet: Ich stehe auf, wälze meine Bücher und gehe dann wieder ins Bett. Das mache ich für die nächsten sieben Jahre und dann bin ich Anwalt.“ Sondern Leute, die sagen „Okay, wir leben in einer modernen Zeit und müssen mit dieser Materie umgehen.“ Das Erste, was ich aktiv machen würde, ist, wie vorhin erwähnt, zu versuchen, praktische Code-Erfahrungen zu sammeln, also einfach mal selbst zu programmieren. Wenn das einem Spaß macht ist das sehr gut und es kann auch eine schöne Ergänzung zum juristischen Arbeiten sein.

Das Legal Tech-Feld spaltet sich auf in Leute, die einfach nur sehr gerne reden und Leute, die selbst aktiv etwas machen. Und wenn man merkt, dass man lieber in die zweite Gruppe gehören möchte, ist man in dem Feld richtig. Es war – gerade aus meiner eigenen Erfahrung – sehr schön zu sehen, dass die Probleme beim Programmieren nicht in einem Meinungsstreit enden, sondern in einem klaren binären Ergebnis: Die Anwendung funktioniert oder sie funktioniert nicht. Und wenn sie nicht funktioniert, liegt der Fehler meistens beim Programmierer. Das ist eine angenehme Erfahrung und man hat sehr viele Erfolgserlebnisse. Zudem kann die Art zu denken, die man aus dem Juristischen schon kennt, auch an Probleme beim Programmieren herangetragen werden.

CTRL: Wir hätten Sie noch nach Literaturempfehlungen gefragt, es hört sich jetzt aber mehr danach an, dass die richtige Herangehensweise „learning by doing“ ist? Hauptsache man interessiert sich dafür, tauscht sich mit anderen Interessierten aus und dann nimmt das seinen Lauf, verstehen wir Sie da richtig?

Ja und nein. Es gibt trotzdem Sachen, die man parallel machen kann. Ich habe jetzt keine klassischen Literaturempfehlungen, weil ich kein Mensch bin, der gerne liest, sondern jemand, der gerne hört oder sieht. Allgemein zum Thema Legal Tech ist der Podcast von Martin Fries und seine Online-Vorlesung "Legal Tech" (dazu hier) super. Der zeigt auf, wie spannend der Bereich Legal Tech eigentlich ist. Ich mache auch mal ein wenig Eigenwerbung: Ich bin Dozent an der LMU München und halte eine Vorlesung „Grundlagen IT & Recht“ (dazu

CS50's Introduction to Computer Science

CS50's Introduction to Computer Science ist ein Kurs der Harvard University, der kostenlos über die Online-Plattform edX abgerufen werden kann. Der Kurs führt seine Teilnehmer, unter der Anleitung von David J. Malan (Gordon McKay Professor of the Practice of Computer Science an der Harvard University) und seiner wissenschaftlichen Mitarbeiter, in die Grundlagen der Informatik und des Programmierens ein. Informatik- oder Programmierkenntnisse werden für eine Teilnahme nicht vorausgesetzt. Der Kurs setzt sich zusammen aus einer Online-Vorlesung und praktischen Programmieraufgaben zu deren Lösung die Programmiersprachen C, Python und SQL angewendet werden. Diese sogenannten „problem sets“ erfordern die eigenständige Lösung von real existierenden Problemen, z.B. der Programmierung einer Anwendung, die überprüft, ob eine Kreditkartennummer gefälscht ist oder nicht. Dabei lernt man grundlegende Konzepte des Programmierens. Insbesondere die Anwendung der Programmiersprache C verlangt, angesichts ihrer wenigen selbst ausführenden Elemente, ein eigenständiges Formulieren des Programmcodes und führt damit zu einem tiefergehenden Verständnis einzelner Programmierkonzepte.

hier), in der ich mir technische Grundlagen anschau und den Teilnehmern vermittele. Ein klasse Kurs um nur Technik und Programmierung zu lernen ist „CS50's Introduction to Computer Science“ der Harvard University, der kostenlos über die Online-Lernplattform edX abgerufen werden kann (siehe Infobox). Der Kurs ist umwerfend gut.

CTRL: Zum Abschluss: Was würden Sie einer Jurastudentin oder einem Jurastudenten mit auf den Weg geben?

Auch wenn man mit Blick auf das Examen wahnsinnig viel Arbeitsbelastung hat, lohnt es sich die Augen offenzuhalten für Neues und für das, was einem besonders Spaß macht.

Vielen Dank für das Gespräch.

Das Interview führten Clarissa Kupfermann und Louis Goral-Wood



Clarissa studiert Jura an der Universität zu Köln und arbeitet als studentische Hilfskraft am dortigen Institut für Straf- und Strafprozessrecht.



Louis studiert Jura an der Universität zu Köln. Neben seinem Studium arbeitet er als studentischer Mitarbeiter am Institut der deutschen Wirtschaft (Kompetenzfeld: Bildung, Zuwanderung und Innovation).