

Mathias Knoll, Michael Brickmann, Elmar Krainz (Hg.)

Cloud Computing Lab

Sammelband

gefördert
von der



FFG
Forschung wirkt.

IT+

FH | JOANNEUM
University of Applied Sciences



© 2025

DI(FH) Mathias Knoll MSc, DI(FH) Michael Brickmann MA,
FH-Prof. DI Dr. Elmar Krainz; Harald Schwab BSc MSc,
Priska Steininger BSc MSc MBA, Stefanie Wieser BSc MA,
Ing. Stefan Rappl BSc, FH-Prof. Mag. Dr. Sabine Proßnegg LL.M., FH-
Prof. Mag. Dr. Kaja Unger



CC BY 4.0 2025 by DI(FH) Mathias Knoll MSc, DI(FH) Michael Brickmann MA, FH-Prof. DI Dr. Elmar Krainz; Harald Schwab BSc MSc, Priska Steininger BSc MSc MBA, Stefanie Wieser BSc MA, Ing. Stefan Rappl BSc, FH-Prof. Mag. Dr. Sabine Proßnegg LL.M., Prof. Mag. Dr. Kaja Unger

Dieses Werk ist lizenziert unter der Creative Commons Attribution 4.0 Lizenz (BY). Diese Lizenz erlaubt unter Voraussetzung der Namensnennung der Urheberin die Bearbeitung, Vervielfältigung und Verbreitung des Materials in jedem Format oder Medium für beliebige Zwecke, auch kommerziell. (Lizenztext: <https://creativecommons.org/licenses/by/4.0/deed.de>)

Die Bedingungen der Creative-Commons-Lizenz gelten nur für Originalmaterial. Die Wiederverwendung von Material aus anderen Quellen (gekennzeichnet mit Quellenangabe) wie z.B. Schaubilder, Abbildungen, Fotos und Textauszüge erfordert ggf. weitere Nutzungsgenehmigungen durch den jeweiligen Rechteinhaber.

Verlag der FH JOANNEUM Gesellschaft mbH

Alte Poststraße 149

A-8020 Graz

www.fh-joanneum.at

ISBN 978-3-903318-64-9

eISBN 978-3-903318-65-6

Umschlagbild:	DI (FH) Mathias Knoll, MSc
Layout:	Mag. (FH) Alexandra Gößlbauer, MA DI (FH) Mathias Knoll, MSc
Druck:	Druckhaus Kurz GmbH, A-8680 Mürzzuschlag

Cloud Computing Lab

¹

<https://cola.fh-joanneum.at>

Sammelband

Gefördert von der



2

Herausgegeben von

Mathias Knoll, Michael Brickmann, Elmar Krainz

¹ Schriftarten: „Loki Cola“ (<https://www.dafont.com/de/loki-cola.font>) und „Colagraph“ (<https://www.dafont.com/de/colagraph.font>)

² Die Österreichische Forschungsförderungsgesellschaft (<https://www.ffg.at/>)

Vorwort

Elmar Krainz

Wir leben in einer Zeit, in der digitale Technologien nicht nur unseren Alltag bestimmen, sondern auch unsere Arbeitswelt, unsere Kommunikation und unser gesellschaftliches Miteinander prägen. Informatik ist dabei weit mehr als ein technisches Fach: Sie ist eine Denkweise, ein Werkzeug und eine kreative Ausdrucksform. Wer Informatik versteht, versteht die Sprache der Zukunft. Wer sie anwenden kann, ist nicht nur Nutzer:in digitaler Anwendungen, sondern Gestalter:in einer Welt, die zunehmend durch Software, Netzwerke und intelligente Systeme geprägt wird.

Digitale Produkte und Services entstehen nicht zufällig – sie beruhen auf sorgfältig geplanter und entwickelter Informationstechnologie. Durch kontinuierliche Innovationen in der Softwareentwicklung schaffen wir effizientere Arbeitsabläufe, neue Formen der Zusammenarbeit, technische Hilfsmittel im Alltag und Lösungen für gesellschaftliche Herausforderungen. Informatik ermöglicht es uns, große Datenmengen zu analysieren, Menschen über Kontinente hinweg zu vernetzen und komplexe Probleme in nahezu allen Lebensbereichen zu verstehen und zu bewältigen. Doch obwohl digitale Technologien selbstverständlich genutzt werden, bleibt ihr Innenleben für viele verborgen.

Genau hier setzt die durch die Österreichische Förderungsgesellschaft FFG finanzierte CoLa-Projektreihe der FH JOANNEUM an. Mit dem Projekt **CoLa**, dem „**Coding Lab**“ wurde der Grundstein gelegt, um Schülerinnen und Schüler frühzeitig an Softwareentwicklung heranzuführen. In einer Welt voller Apps, Geräte und digitaler Dienste stellt sich die Frage: Warum sollte man wissen, wie Software entsteht, wenn man sie auch einfach benutzen kann? Die Antwort ist einfach: Programmieren ist ein

kreativer Prozess. Es erlaubt, Ideen auszuprobieren, Probleme zu lösen und Neues zu erschaffen. Die Jugendlichen lernten, dass Softwareentwicklung weit mehr umfasst als das Schreiben von Code. Sie erfuhren, wie aus einer ersten Idee – über Planung und Umsetzung – ein funktionierendes digitales Produkt entsteht. Durch altersgerechte Workshops wurden logisches Denken und Kreativität gefördert, und im Coding Lab der FH JOANNEUM in Kapfenberg entstand ein Raum, der junge Menschen begeistert, inspiriert und ihnen zeigt, wie viel Potenzial in ihnen steckt.

Das darauffolgende Projekt **CoLa 2.0**, das „**Cloud Computing Lab**“ öffnete den Blick für die nächste Ebene moderner Informatik: die Cloud. Sie ist das technische Rückgrat unseres digitalen Lebens – von sozialen Medien über Streamingdienste bis zu global operierenden Unternehmen. In den Workshops wurde Cloud Computing verständlich und greifbar gemacht. Themen wie Netzwerke, Server, Datenbanken oder verteilte Systeme, die oft als abstrakt empfunden werden, wurden anschaulich erklärt und erlebbar gemacht. So entstand ein neues Verständnis dafür, wie moderne IT-Infrastrukturen funktionieren und welche beruflichen Möglichkeiten sich daraus ergeben. Besonderer Wert wurde daraufgelegt, auch Jugendliche mit Migrationshintergrund einzubeziehen, um Chancengleichheit zu fördern und der Region Obersteiermark-Ost langfristig neue Impulse im technisch-naturwissenschaftlichen Bereich zu geben.

Mit dem geplanten Projekt **CoLa 3.0** – „**COol Kids Learn Ai**“ richtet sich der Blick nun auf eines der bedeutendsten Zukunftsthemen: künstliche Intelligenz. KI ist längst Bestandteil unseres täglichen Lebens – doch ihre Funktionsweise bleibt für viele unklar. Dabei ist es gerade für junge Menschen entscheidend, KI nicht nur als Werkzeug zu nutzen, sondern zu verstehen, wie sie funktioniert, wie sie Entscheidungen trifft und welche gesellschaftlichen Folgen ihr Einsatz hat. Im Rahmen von interaktiven Rollen- und Planspielen sowie kreativen Workshops sollen Schüler:innen die grundlegenden Konzepte der KI kennenlernen. Durch einfache, aber wirkungsvolle Beispiele wird vermittelt, wie Maschinen lernen, wie

Algorithmen aufgebaut sind und wie KI-Systeme in verschiedensten Bereichen eingesetzt werden. Gleichzeitig werden Lehrpersonen durch gezielte Weiterbildungen befähigt, das Thema künstliche Intelligenz langfristig in den Unterricht einzubinden und junge Menschen auch bei wissenschaftlichen Arbeiten zu diesem Thema zu begleiten.

Alle drei Phasen der CoLa-Reihe verfolgen ein gemeinsames Ziel: Kinder und Jugendliche frühzeitig für Informatik zu begeistern, ihnen zentrale digitale Kompetenzen zu vermitteln und ihnen zu zeigen, dass sie die digitale Zukunft mitgestalten können. Informatik ist eine universelle Sprache, die unabhängig von Herkunft, Geschlecht oder sozialem Hintergrund erlernt und verstanden werden kann. Sie eröffnet Chancen, stärkt Selbstvertrauen und fördert genau jene Fähigkeiten, die in einer modernen Gesellschaft unverzichtbar sind: logisches Denken, Kreativität, Teamarbeit und Problemlösungskompetenz.

So leisten die Projekte CoLa, CoLa 2.0 und auch das geplante CoLa 3.0 einen wichtigen Beitrag für die Region, für das Bildungssystem und vor allem für die jungen Menschen, die unsere Zukunft gestalten werden. Denn wer die Welt von morgen verstehen will, muss die Technologien von heute begreifen – und wer sie begreift, kann sie verändern. Genau dazu wollen wir motivieren.

Inhaltsverzeichnis

Coding Lab	1
<i>Elmar Krainz</i>	
Einleitung	1
Ziele	3
Umsetzung	3
Wirkung	4
Cloud Computing Lab	6
<i>Mathias Knoll</i>	
Einleitung	6
Cloud Computing Matrix	8
Workshops	11
Netzwerke	15
<i>Mathias Knoll</i>	
Materialien	15
Basisspiel (Primarstufe)	17
Netzwerkaufbau	22
Aufgaben	25
Erweiterung 1 (Sekundarstufe 1)	26
Erweiterung 2 (Sekundarstufe 2)	35
Technischer Kontext	39
Zusammenfassung	40
Server	41
<i>Harald Schwab</i>	
Was ist ein Server?	42
Was macht ein Server	42
Wie kommuniziert man mit einem Server	43

Arten von Servern im Netzwerk.....	45
Material.....	50
Basisspiel (Primarstufe)	51
Erweiterung 1 (Sekundarstufe 1).....	56
Erweiterung 2 (Sekundarstufe 2)	58
Speicher	59
<i>Michael Brickmann, Mathias Knoll</i>	
Bits (binary digits) & Bitfolgen	60
Transistoren.....	61
Flip-Flop Speicherung eines Bits	62
Datenbanken	66
<i>Priska Steininger</i>	
Dinge (Entitäten).....	68
Datenmodellierung, ER-Modell und Relationen.....	71
Basisspiel (Primarstufe)	74
Erweiterung 1 (Sekundarstufe 1).....	75
Erweiterung 2 (Sekundarstufe 2)	80
Apps & Services	85
<i>Stefan Rappl, Harald Schwab, Stefanie Wieser</i>	
Fragen API (Sekundarstufe 2)	86
Quiz App (Sekundarstufe 2)	91
Block API (Sekundarstufe 2)	96
Safer Internet & Cloud Sicherheit im digitalen Raum	97
<i>Sabine Proßnegg, Kaja Unger</i>	
„Ge Cloud“.....	97
Selbstbewusstsein und Offenheit	102
Conclusio	103
Weiterführende Informationen:	104

Zusammenfassung	105
<i>Mathias Knoll</i>	
Glossar	108
Literaturverzeichnis	111
Abbildungsverzeichnis	114
Tabellenverzeichnis.....	118

Alle Unterlagen und weiterführende Informationen findet man
ebenfalls unter:



<https://cola.fh-joanneum.at>

Coding Lab

Elmar Krainz

Kinder wachsen heute selbstverständlich mit digitalen Geräten und Diensten auf. Das Nutzen digitaler Technologien gehört zum Alltag – deren Entwicklung jedoch nicht. Bildungseinrichtungen stehen vor der Aufgabe, junge Menschen auf eine Zukunft vorzubereiten, in der digitale Kompetenzen zentral sind. Obwohl zahlreiche Initiativen wie code.org oder „Scratch“¹ Kinder bereits früh für das Programmieren begeistern, beschränken sich viele Angebote auf blockbasiertes Programmieren und Vermitteln nicht den gesamten Entwicklungsprozess digitaler Produkte.

Das Projekt CoLa – Coding Lab verfolgt daher einen erweiterten Ansatz: Kinder erhalten einen altersgerechten Einblick in alle Phasen des Softwareentwicklungsprozesses – von der Ideenfindung bis zur Veröffentlichung. Das Coding Lab unterstützt Schulen vor Ort, bietet eine inspirierende Lernumgebung und ermöglicht textbasiertes Programmieren über eine eigens entwickelte, kindgerechte Entwicklungsumgebung. Über 500 Kinder haben das Coding Lab genutzt und erste Erfahrungen in Softwareentwicklung gesammelt (Krainz, et al., 2022)

Einleitung

Digitale Technologien prägen alle Lebensbereiche. Jugendliche nutzen Smartphones, Tablets und Computer intuitiv, doch nur wenige wissen, wie digitale Produkte entstehen. Gleichzeitig steigt der Bedarf an IT-Fachkräften deutlich – allein in Österreich fehlen laut (Schneider, et al., 2020) rund 24.000 Spezialist:innen. Umso wichtiger ist es, die nächste Generation frühzeitig auf ein Leben und Arbeiten in einer digitalisierten Gesellschaft vorzubereiten.

¹ Siehe <https://scratch.mit.edu/>

Das österreichische Bildungsministerium¹ fördert seit Jahren die digitale Transformation des Schulwesens. Neben Infrastrukturmaßnahmen gewinnt die Vermittlung digitaler Kompetenzen zunehmend an Bedeutung.

Der Gedanke, Kinder zu programmieren zu befähigen, ist nicht neu. Erste didaktische Ansätze wie LOGO (Lukas, 1972) oder ToonTalk (Kahn, 1996) sind schon einige Jahrzehnte alt. Mit dem Einzug mobiler Geräte entstanden viele visuelle Programmiersprachen wie Scratch (Maloney, et al., 2010) , Catrobat (Slany, 2012) oder Roberta (Bredenfeld & Leimbach, 2010) , die per Drag-and-drop einfache Programme ermöglichen.

Diese Werkzeuge senken die Einstiegshürde, bilden aber nur einen kleinen Teil realer Softwareentwicklung ab. Daher gewinnen kindgerechte Ansätze für textbasierte Sprachen wie Python (Tabet, et al., 2016) oder JavaScript (Horváth & Menyhárt, 2014) zunehmend an Bedeutung, da sie einen direkteren Bezug zur realen IT-Praxis herstellen.

Das Projekt CoLa – Coding Lab, verfolgt das Ziel, Kindern und Jugendlichen einen spielerischen Zugang zur Informatik zu ermöglichen. Dabei unterscheidet sich CoLa von anderen Initiativen in zwei wesentlichen Punkten:

- Ganzheitlicher Ansatz: Kinder lernen den gesamten Softwareentwicklungsprozess kennen – von der Planung über die Entwicklung bis hin zur Anwendung und Weiterentwicklung.
- Unterstützung regionaler Schulen: Das Coding Lab ist als „Digital Maker Space“ konzipiert, der Schulen, Lehrpersonen und Eltern offensteht. Die Gestaltung des Labors orientiert sich an modernen Start-up-Büros und schafft eine motivierende Lernatmosphäre.

¹ BMB, <https://digitaleschule.gv.at/>

Ziele

Das Projekt verfolgte drei zentrale Zielsetzungen. Erstens der Aufbau eines regionalen Coding Labs. Das Labor dient als kreativer Lernort und digitaler Maker Space. Kinder, Lehrkräfte und Eltern können hier eigene digitale Produkte planen, gestalten und programmieren – sie werden vom Konsumenten („user“) zum Produzenten („prosumer“).

Zweitens die Gestaltung und Umsetzung von Workshops und Kurse für Schulklassen. Die Workshops vermitteln den Softwareentwicklungsprozess altersgerecht: von der Idee zur Planung, der Umsetzung und schließlich der Anwendung. Das Angebot wurde dabei individuell auf die jeweilige Schule abgestimmt. Inhalte reichen von Prototypen über Computational Thinking bis zur App- oder Drohnenprogrammierung. Und drittens die Weiterbildung von Lehrkräften, Lehrpersonen lernen Grundlagen der Softwareentwicklung kennen, um künftig eigenständig Projekte umsetzen und Schüler:innen bei Informatikthemen begleiten zu können.

Umsetzung

Das Coding Lab orientiert sich am klassischen Software Development Lifecycle (SDLC) (Shylesh, 2017) , den das Projekt auf drei Kernphasen reduziert:

- **Planung** (Ideenfindung, Prototypen, Personas)
- **Entwicklung** (App- und Spielentwicklung, agile Methoden)
- **Service** (Marketing, Weiterentwicklung, Geschäftsmodelle)

Für jede Phase wurden altersgerechte Kurse entwickelt.

Ein inspirierender Lernort ist wesentlich für positive Lernerfahrungen. Daher wurde das Coding Lab mit flexiblen Möbeln, Arbeitsinseln, Sofas und höhenverstellbaren Tischen ausgestattet – bewusst anders als klassische Klassenzimmer.

Für die technische Umsetzung kommen Raspberry Pi 400 Geräte zum Einsatz. Sie bieten eine mobile, robuste und kostengünstige Entwicklungsplattform für Schulen.

Im Projekt wurde eine browserbasierte Entwicklungsumgebung auf Basis von p5.js (McCarthy, et al., 2015) geschaffen. Anders als blockbasierte Systeme setzt CoLa bewusst auf textbasiertes Programmieren. Die Erfahrung zeigt: Die Lernkurve ist zu Beginn steiler, aber die Kinder meistern die Herausforderung schnell und gewinnen ein realistischeres Verständnis für Softwareentwicklung.

Wirkung

Das Projekt wurde in Kapfenberg durchgeführt. Die Stadt zählt rund 22.000 Einwohner:innen, davon 20 % im Alter bis 20 Jahre und ebenfalls rund 20% mit Migrationshintergrund – ideale Voraussetzungen für ein breites, diverses Zielpublikum.

Im Projektzeitraum waren 8 Volksschulen sowie 4 weiterführende Schulen am Projekt beteiligt. In 29 Workshops wurden rund 430 Schüler:innen im Rahmen von Besuchen der Partnerschulen erreicht. In der Game-Development-Sommerschule kamen weitere 35 Teilnehmer:innen und in der Kinder-Softwarefirma im Rahmen der „Freitopa“-Initiative

75 Schüler:innen mit dem Thema Coding in Kontakt. Insgesamt konnten durch CoLa 1.0 540 Kinder und Jugendliche zwischen 6 und 18 Jahren einen Einblick in das Thema Softwareentwicklung erhalten.

Die Rückmeldungen der Schulen zeigen deutlich, dass das Projekt eine nachhaltige und positive Wirkung entfaltet: Die Kinder arbeiten mit hoher Motivation, berichten von wertvollen Lernerfahrungen und entwickeln ein spürbar gesteigertes Interesse an IT und Softwareentwicklung. Besonders hervorgehoben wird zudem die modern gestaltete

Lernumgebung, die von Schüler:innen wie Lehrpersonen gleichermaßen geschätzt und als bereichernd für den Unterricht wahrgenommen wird.

Das Projekt CoLa zeigt eindrucksvoll, dass Kinder und Jugendliche durch einen inspirierenden Lernort, praxisnahe Workshops, textbasiertes Arbeiten mit realem Softwarebezug, eine enge Kooperation mit Schulen und einen klar strukturierten Entwicklungsprozess nachhaltig für Softwareentwicklung begeistert werden können. Mit 540 erreichten Kindern wird sichtbar, dass Coding Labs eine wirksame Antwort auf die zentralen Herausforderungen digitaler Bildung darstellen.

Die Weiterentwicklung umfasst den Aufbau eines virtuellen Coding Labs, selbstgeführte Online-Lerneinheiten, eine vertiefte Auseinandersetzung mit IT-Sicherheit, Infrastruktur und Datenschutz, erweiterte Angebote für Lehrpersonen sowie die Integration von Cloud- und KI-Themen in den Folgeprojekten CoLa 2.0 und CoLa 3.0. Damit entsteht ein Bildungsraum, in dem junge Menschen digitale Technologien nicht nur nutzen, sondern verstehen und aktiv mitgestalten.

Cloud Computing Lab

Mathias Knoll

Das Projekt CoLa 2.0 - Cloud Computing Lab ist die Fortsetzung des Computing Lab und inkorporiert um Cloud ergänzte Workshops und Vortragsreihen. Es stellt dabei eine neue Sammlung von verschiedenen Methoden, Programmen sowie Apps zur Verfügung und bietet pädagogisch aufbereitete Wissensvermittlung. Im Rahmen von Rollenspielen und Workshops (Cloud Labs) mit Schulen werden das logische Denkvermögen und die Kreativität von Schüler:innen durch eine altersgerechte Aufbereitung gefördert. Dadurch eröffnet sich ein neuer Zugang zur Technologie „Cloud Computing“, der von der Lebenswelt der Kinder und Jugendlichen ausgeht.

Einleitung

Täglich sind wir von Informationstechnologien umgeben. Viele Online-Angebote werden von uns genutzt. Davon sehen wir jedoch nur die Spitze des Eisberges. Darunter liegt ein Geflecht aus Technologien und Firmen. Die wichtigste technische Errungenschaft sind Computernetzwerke und das globale Netzwerk namens „Internet“.

Oft wird der Begriff „Cloud“ als Synonym für jenes Internet genannt. Die „Cloud“ umfasst jedoch viel mehr. Sie steht metaphorisch für eine globale Infrastruktur, ein globales Netzwerk und den Diensten darin. Sie ist kein physisch fassbares Gebilde, sondern ein über die ganze Welt verteiltes und miteinander verbundenes Ökosystem an Diensten und Ressourcen. Das Symbol einer Wolke illustriert Ortsunabhängigkeit und Beweglichkeit. Einfach erklärt steckt hinter dem Begriff „Cloud Computing“ die Verarbeitung von Daten über das Internet. Zahlreiche altbekannte Technologien werden mit dem Ziel eingesetzt, Daten dezentral zu speichern,

ihre Kapazitäten flexibel zu provisionieren (d.h. eine flexible Einrichtung einer IT- Infrastruktur mit virtueller Hardware und Diensten) und den Zugang zu diesen immer und überall verfügbar zu machen.

Klassische Teile eines Softwareprojekts, wie Datenbank oder Benutzerverwaltung können so an einen Cloud-Anbieter ausgelagert werden, um mehr Entwicklungszeit auf die Kerngebiete einer Anwendung zu verwenden. Das Einbinden von Cloud-Diensten ist einfach und beliebig erweiterbar. Heute verwenden vor allem Handy-Apps Dienste namhafter Anbieter wie Microsoft, Google oder Amazon. Benutzer:innen wissen meistens nicht Bescheid, welche Drittanbieter hinter einer App stehen und welche Technologien im Hintergrund arbeiten.

Das von der Österreichischen Forschungsförderungsgesellschaft (FFG) (FFG, 2024) geförderte Projekt „Cloud Computing Lab“ (CoLa 2.0) (Knoll & Wieser, 2025) widmet sich der Vermittlung von Hintergrundwissen zu den Grundlagen der Cloud Technologien, aber auch den Implikationen derer Verwendung. Im Rahmen des Projekts wurden zahlreiche Workshops durchgeführt. Ausgerollt wurden diese auf Schulen in der Primär- und Sekundarstufe 1 und 2. Insgesamt bekamen über 700 Schüler:innen einen Einblick in den Bereich der Cloud Technologien.

Ausgehend vom Überblick, des amerikanischen „National Institute of Standards and Technology“ (NIST) (Mell & Grance, 2011) kann man Cloud Computing in folgende Bereiche aufteilen:

- **Netzwerke**

Kommunikation in Computernetzwerken. Technische Lösung und Konzept von Netzwerken und Routing. Einblick auf die Geräte, Funktionsweisen und Architekturen.

- **Server**

Client-Server Beziehung in Netzwerken und Kommunikation zwischen Partner:innen in der Anwendungsschicht. Einblick auf gängige

Protokolle wie HTTP (Hypertext Transfer Protocol), DNS (Domain Name System) und DHCP (Dynamic Host Configuration Protocol) , welche grundlegende Dienste im Internet und in Netzwerken sind. (Tanenbaum, et al., 2024)

- **Speicher**

Wie werden Daten gespeichert? Wie sehen Daten auf der Festplatte aus? Lokaler oder verteilter Speicher, Repräsentation der Daten auf Speichermedien. Wie kommen Daten in/aus der Cloud?

- **Datenbanken**

Konzepte hinter der Speicherung von Informationen, Relationales und noSQL Datenbankkonzept. Fortführend: „Big Data“ in der Cloud.

- **Anwendungen**

Anwendungen in der Cloud und Funktionalitäten in der Cloud, die man in eigene Anwendungen bringen kann (Beispiel wäre u.a. die Benutzerverwaltung)

- **Dienste**

APIs (Application Programming Interfaces) und, oder als Web-Services ausgeführt.

Cloud Computing Matrix

Auf diesen Überblick an Technologien baut die Cloud-Computing-Lab-Matrix (siehe Abbildung 1), welche die Grundlagen für die beteiligten Schüler:innen verständlich macht, auf. Die Vermittlung der einzelnen Unterpunkte erfolgt mit alters- bzw. schulstufengerechten Methoden. Die Zuordnung, welche Inhalte in welcher Schulstufe vermittelt werden können, sind aus der im Folgenden dargestellten Matrix abzulesen. In den

Zeilen der Matrix sind die groben Inhalte der einzelnen Module angeführt, und in den Spalten die Schulstufen.

Die Zielgruppe für regelmäßige Workshops außerhalb schulischer Aktivitäten sind regional ansässige Schüler:innen. Die Kinder und Jugendlichen – insbesondere auch Mädchen sowie Schüler:innen mit Migrationshintergrund sollen für vertiefende IT-Ausbildungen begeistert werden, um längerfristig dem Mangel an IT-Fachkräften entgegenzuwirken und das MINT Mindset an Schulen weiter zu fördern.

Cloud Computing Lab Matrix			Schulstufen		
			Primärstufe	Sekundarstufe I	Sekundarstufe II
Technologebereiche laut NIST Definition des Cloud Computings	Netzwerke	Netzwerkgrundlagen Geräte, Konfiguration, Funktionen	X	X	X
		Konzept eines Schichtenmodells OSI (Open Systems Interconnect)		X	X
		Client-Server Architektur (Socket Verbindung)			X
	Server	HTTP - Webserver Hypertext Transfer Protocol	X	X	X
		DNS – Namensdienst im Internet Domain Name System		X	X
		DHCP – Hostkonfiguration Dynamic Host Configuration P.		X	X
	Speicher	Grundlagen der Datenspeicherung Mechanisch, Flip-Flop Schaltung, ...	X	X	X
		Datenschutz und -sicherheit		X	X
	Datenbanken	Strukturierung von Information (Entity-Relationship Modell)	X	X	X
		Datenbankabfragen (Structured Query Language)		X	X
		Fortführende Konzepte (BigData und noSQL)			X
	Apps	Apps mit Cloud Anbindung (Firebase, Heroku, ...)		X	X
		Privatsphäre (z.B. Facebook als Cloud App)		X	X
	Dienste	Dienste im Web (Cloud Ressourcen im Web)	X	X	X
		Web-API (Representational State Transfer)			X

Abbildung 1: Cloud Computing Lab Matrix, © Knoll.

Die größte Herausforderung an das Projekt sind die im Cloud Computing ineinander komplex verwobenen Technologien in eine für Schüler:innen verständliche Form zu bringen. Bei Handy-Apps muss zum Beispiel auf Daten aus dem Internet zugegriffen werden. Hier spielen Netzwerktechnologien, Webengineering und mobile Plattformen zusammen, um Produktkataloge, Termine oder Dokumente über eine Datenbank aus dem Internet zu beziehen. „Rich Internet Applications“, welche umfangreiche Funktionalitäten am Browser anbieten, wie zum Beispiel Microsoft mit der Office 365 oder Google mit Online-Werkzeugen wie Docs kombinieren Funktionalitäten des Browsers mit Daten von Webservern. Die Cloud kann Speicherplatz, Rechenkapazität oder Funktionalität zur Verfügung stellen. Cloud-Dienste sollen einfachen Zugang über das Netzwerk, üblicherweise das Internet bieten. Auch das bedarfsgerechte und einfache Anpassen der Online-Ressourcen an das eigene Software-Projekt spielt eine große Rolle.

Die altersgerechte Aufbereitung der Basistechnologien und die Erarbeitung derer Funktionsweisen im Rahmen von Workshops mit den beteiligten Schulen soll das logische Denkvermögen und die Kreativität im Umgang mit Cloud-basierten Inhalten fördern. Exkursionen zu Firmen, die mit der Cloud arbeiten und selbst in der Cloud sind, sollen das Konzept fassbarer machen. Da man in der Cloud die Dienste von Drittanbietern verwendet soll das Bewusstsein ebenso in Hinblick auf Datenschutz, -sicherheit und Privatsphäre geschärft werden.

Exkursionen zum Partner KNAPP Systemintegration GmbH (KSI) in Leoben, welcher in seinen innovativen Logistiklösungen Cloud-Dienste sowohl im Einsatz hat als auch selbst betreibt und hostet, sollen den Schüler:innen Anwendungsgebiete und die Technik hinter Cloud Computing näherbringen.

Workshops

Um eine größere Anzahl von Schulen und Schüler:innen zu erreichen, wurden sämtliche Workshops über den Webauftritt des Projekts veröffentlicht (Knoll, 2025). Neben einer Präsentation und einer Broschüre für Schüler:innen begleitet eine erweiterte Fassung Pädagog:innen durch den Workshop und erklärt Hintergründe und den realen Bezug. Etwaiges Material wird auf der Homepage des Projekts ebenfalls angeboten.

Auf Basis der vorgestellten Matrix wurde je ein Workshop zu jeder Technologie entwickelt. Analog zu den Schulstufen wurden die Inhalte dieses Workshops in drei aufeinanderfolgende Bereiche mit steigendem Schwierigkeitsgrad aufgeteilt. Indikatoren in den Workshopbeschreibungen teilen die Zielgruppe/Schwierigkeit mit:



Abbildung 2: Stufenweiser Workshop Aufbau, © Knoll.

Die Kapitel zu den Workshops sind mit Pfeilen gekennzeichnet, die den jeweiligen Schwierigkeitsgrad anzeigen:

- Basis: Primarstufe
- Erweiterung 1: Sekundarstufe I
- Erweiterung 2: Sekundarstufe II

Diese Pfeile orientieren sich an der im Text beschriebenen Matrix und verdeutlichen, für welche Zielgruppe bzw. welchen Schwierigkeitsgrad die Workshop-Inhalte vorgesehen sind.

„Die Cloud ist mehr das Wie und nicht mehr das Wo.“

abgewandelt von “Cloud is about how you do computing, not where you do computing.”

Paul Maritz, VMware CEO

In Workshops mit beteiligten Bildungseinrichtungen werden das analytische Denken und die Innovationsfähigkeit von Schüler:innen durch die altersgerechte Vermittlung von Cloud-Technologien gestärkt. Hier lernen sie, wie Cloud-Infrastrukturen aufgebaut und effizient genutzt werden können. Ziel des Projekts ist es, Kinder und Jugendliche in der Region, um Kapfenberg für den Bereich des Cloud-Computings zu begeistern und ihnen die technischen sowie kreativen Fähigkeiten zu vermitteln, die in der modernen IT-Welt erforderlich sind.

Über die Website „<https://cola.fh-joanneum.at>“ bietet das Projekt nicht nur für seine Partner:innen sondern für alle interessierte Schulen Workshops und Online-Apps (siehe Abbildung 3) an. Über Kooperationszuschüsse können Schulen Mittel bewilligt werden, um eigene Projekte im Cloud-Umfeld auf die Beine zu stellen.

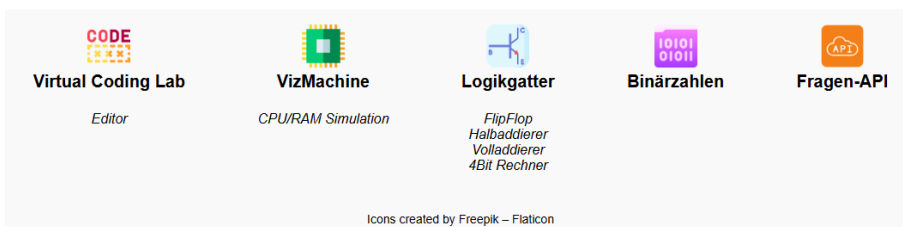


Abbildung 3: Entwickelte Apps auf der Site, © Knoll, FlatIcon.com

Alle im Projekt verwendeten Materialien stehen auch über das Projektende hinaus online zur Verfügung. So können die Inhalte jederzeit erneut abgerufen, vertieft oder eigenständig weiterbearbeitet werden. Darüber hinaus wird die Plattform weiterhin gepflegt und durch regelmäßige Workshops ergänzt, sodass eine lebendige, nachhaltige Lernumgebung bestehen bleibt.

Wir hoffen, dass dieses Projekt nicht nur Wissen vermittelt, sondern auch Neugier und Begeisterung für die digitale Welt geweckt hat – und freuen uns darauf, wenn die Reise im Cloud-Universum von der FH JOANNEUM begleitet werden kann.

Cloud Computing Lab

Workshops

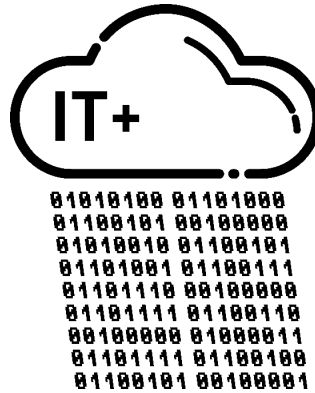


Abbildung 4 Workshop Sujet © Knoll

Die hier dargestellten Workshops geben einen Streifzug durch die Inhalte und Methoden. Auf der Homepage findet man jedoch weit umfangreicheres Material, ergänzende Informationen und verschiedene digitale Tools, die die Umsetzung und Vertiefung der Themen erleichtern.

Über den QR-Code gelangt man direkt zu diesem erweiterten Angebot auf der Website über <https://cola.fh-joanneum.at>:



Netzwerke

Mathias Knoll

Spielerisch wird in einem Workshop vermittelt, wie Computer miteinander kommunizieren und wie Nachrichten in einem Computernetzwerk gesendet und empfangen werden. Ein Netzwerkspiel veranschaulicht in vereinfachter Form die Technik, die auch beim Surfen im Internet, beim Versenden von E-Mails oder beim Streamen von Filmen zum Einsatz kommt.

Dieser Workshop deckt diesen Bereich der Matrix ab:

Tabelle 1 Cloud Matrix: Netzwerke, © Knoll

Netzwerke	Netzwerkgrundlagen Geräte, Konfiguration, Funktionen	X	X	X
	Konzept eines Schichtenmodells OSI (Open Systems Interconnect)		X	X
	Client-Server Architektur (Socket Verbindung)			X

Materialien

Das Netzwerkspiel kann mit einfachen, bereits vorhandenen oder ausgemusterten Netzwerkkomponenten durchgeführt werden. So lässt sich alte Hardware kostengünstig und nachhaltig weiterverwenden. Im nebenstehenden Bild sind alle benötigten Geräte und Materialien übersichtlich aufgelistet – von Routern über Kabel bis zu Endgeräten – und zeigen, wie leicht sich das Setup mit minimalem Aufwand umsetzen lässt.

Zum weiteren Material gehören neben der Hardware auch Ausdrucke und Anleitungen, Briefumschläge, Etiketten sowie ergänzende Unterlagen,

die für den Ablauf des Spiels benötigt werden. Diese Materialien unterstützen den praktischen Aufbau und erleichtern die Durchführung des Workshops auf anschauliche und strukturierte Weise.

Hardware zum physischen Zusammenbau des Netzwerks

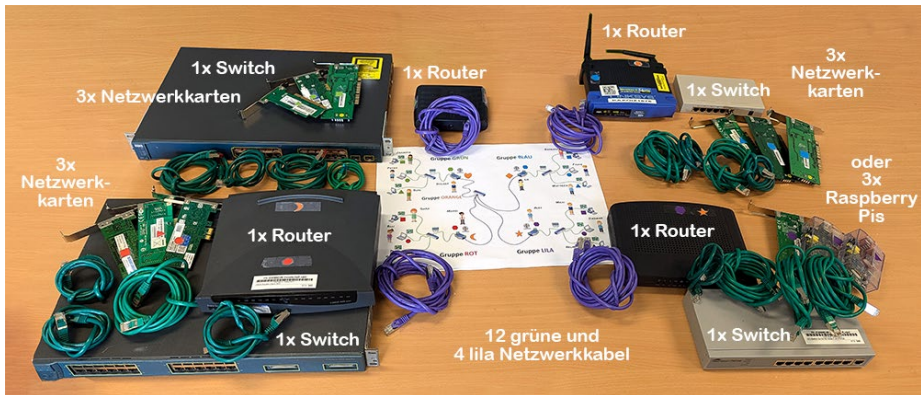


Abbildung 5: Geräte, © Knoll.

Ausdrucke, Anleitungen und Verbrauchsmaterial:



Abbildung 6: Ausdrucke, Anleitungen & Verbrauchsmaterial, © Knoll.

Viele Briefumschläge in drei Größen:



Abbildung 7: Briefumschläge in drei Größen, © Knoll.

Basisspiel (Primarstufe)

Im Netzwerkspiel hat jede Person die Rolle eines Geräts in einem Computernetzwerk. All diese Geräte haben einen Steckplatz für ein Netzkabel. Jedes Gerät hat eine bestimmte Aufgabe.

Es gibt drei Arten von Netzwerkgeräten, die in diesem Spiel verwendet werden und welchen Personen zugeordnet werden können:

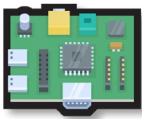


Abbildung 8 Symbol Netzkarte - Flaticon.com

Netzkarten

(Network Interface Cards, NICs) in einem Computer

Eine Netzkarte ist für das Empfangen und Senden von Nachrichten da. Jede Netzkarte erkennt man an ihrem eindeutigen Symbol, welches auf die Karte geklebt wurde:



Abbildung 9: Symbole, eindeutige Markierungen von Netzwerkkarten © Knoll

Mit Symbol ist die so genannte Media Access Control (MAC) Adresse ¹gemeint. Diese ist eine 48bit große Hardwarekennung. Sie wird in Hexadezimal geschrieben und besteht üblicherweise aus einem Teil Herstellerkennung und einem Teil fortlaufender Nummerierung durch den Hersteller. Beispiel: **00-60-2F-AE-FD-7E** (00-60-2F ist die Kennung von Cisco Systems)

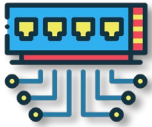


Abbildung 10 Symbol Switch - Flaticon.com

Schalter

(Switch) mit vielen Steckplätzen

Ein Schalter verbindet Geräte in einem Netzwerk. Er hat dafür viele Steckplätze und merkt, wer an ihm angeschlossen ist anhand des „Symbols“, der Kennung der Netzwerkkarte.

Die Funktionalität eines Switches² wird in diesem Spiel vernachlässigt. In dem Spiel leitet er einfach die Nachrichten richtig im Netzwerk weiter. In Wirklichkeit ist dieses Gerät intelligent und merkt sich die Hardwarekennung der an ihm angeschlossenen Geräte, speichert sie und sendet so zielgerichtet weiter:

¹ Siehe <https://de.wikipedia.org/wiki/MAC-Adresse>

² Siehe <https://de.wikipedia.org/wiki/Switch>

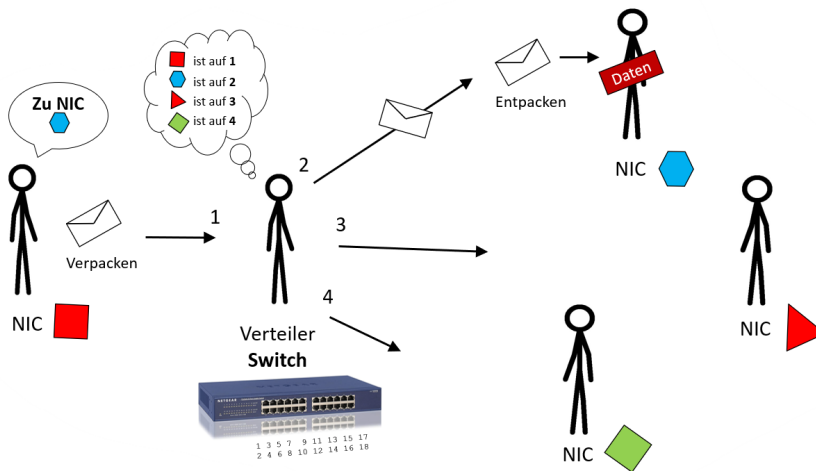


Abbildung 11: Beispiel für die Kernfunktionalität eines Switchs @Knoll

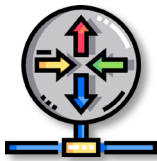


Abbildung 12 Symbol Router - Flaticon.com

Vermittler

(Router) mit mindestens 2 Netzwerkkarten

Ein Vermittler verteilt Nachrichten an andere Netzwerke (zu anderen Gruppen). Er ist mit einer Netzwerkkarte in dem einen mit der anderen Netzwerkkarte in einem anderen Netzwerk ausgestattet. Er weiß, wohin er Nachrichten in andere Netzwerke (zu anderen Gruppen)senden muss.

Die Router kommunizieren untereinander über sogenannte Routingprotokolle. Über diese wissen alle, welches Paket wo hinsoll. Das heißt, dass jeder Router über die gesamte Netzwerktopologie Bescheid weiß.

In unserem kleinen LAN (Local Area Network) wird ein einfaches Routingprotokoll verwendet – das Routing Information Protokoll (RIP)¹. Wenn alle Router sämtliche Information zu allen Netzwerken haben, haben sie die so genannte „Konvergenz“ erreicht:

Wenn man sich das komplette Diagramm des gesamten Netzwerks mit Namen, Gruppen und Symbolen ansieht, dann speichern die 4 Router in diesem Netzwerk folgende Informationen:

(Bitte gegebenenfalls später auf die folgende Seite zurückkommen, wenn das gesamte Netzwerk vorgestellt wurde!)

¹ Siehe https://de.wikipedia.org/wiki/Routing_Information_Protocol



Router Holger Gruppe GRÜN		
<i>In welches Netzwerk will ich?</i>	<i>Über welchen Router erreichbar?</i>	<i>Wie viele Hops?</i>
Gruppe ORANGE	Selbst verbunden	0
Gruppe GRÜN	Selbst verbunden	0
Gruppe Blau	Liz	1
Gruppe Rot	Mario	1
Gruppe Lila	Alex	1



Router Liz Gruppe Blau		
<i>In welches Netzwerk will ich?</i>	<i>Über welchen Router erreichbar?</i>	<i>Wie viele Hops?</i>
Gruppe ORANGE	Selbst verbunden	0
Gruppe GRÜN	Holger	1
Gruppe Blau	Selbst verbunden	0
Gruppe Rot	Mario	1
Gruppe Lila	Alex	1



Router Mario Gruppe Rot		
<i>In welches Netzwerk will ich?</i>	<i>Über welchen Router erreichbar?</i>	<i>Wie viele Hops?</i>
Gruppe ORANGE	Selbst verbunden	0
Gruppe GRÜN	Holger	1
Gruppe Blau	Liz	1
Gruppe Rot	Selbst verbunden	0
Gruppe Lila	Alex	1



Router Alex Gruppe Lila		
<i>In welches Netzwerk will ich?</i>	<i>Über welchen Router erreichbar?</i>	<i>Wie viele Hops?</i>
Gruppe ORANGE	Selbst verbunden	0
Gruppe GRÜN	Holger	1
Gruppe Blau	Liz	1
Gruppe Rot	Mario	1
Gruppe Lila	Selbst verbunden	0

Netzwerkaufbau

Zu Beginn des Workshops werden Computernetzwerke vorgestellt und die elementaren Funktionalitäten der drei wichtigsten Komponenten eines Computernetzwerks erläutert.

Im folgenden Planspiel werden vier Gruppen gebildet. Jede Gruppe bildet ein eigenes Netzwerk mit den dazu notwendigen Netzwerkgeräten. Jede Person spielt die Rolle eines Geräts im Netzwerk:

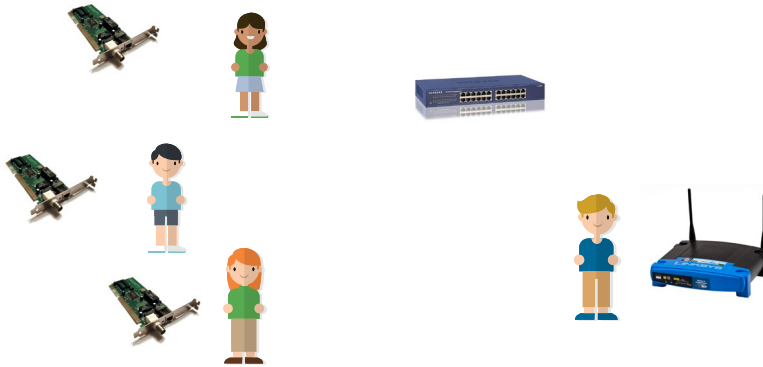


Abbildung 14: Rollenzuteilung: Gerät zu Person © Knoll, PixaBay.com

Ein Computernetzwerk besteht aus verbundenen Netzwerkkarten. Das Planspiel ist auf mindestens 2 Personen pro Gruppe ausgelegt. Die optimale Größe wären 4-5 Personen, wobei 1-2 Personen die Rolle des Vermittlers (Router) übernehmen.

Die Gruppe kann nun die Verkabelung durchführen. Die Farbe der Kabel haben nichts mit deren Eigenschaften zu tun- sie sind nur für einfache Unterscheidbarkeit gefärbt.

„Jeder spielt eine Rolle im Netzwerk“

In folgendem Beispiel sieht man die grüne Gruppe. Die Netzwerkkarten in den Geräten erkennen sich nur an den Symbolen. Alle Netzwerkkarten mit einem grünen Symbol sind in einem physisch verbundenem Netzwerksegment:

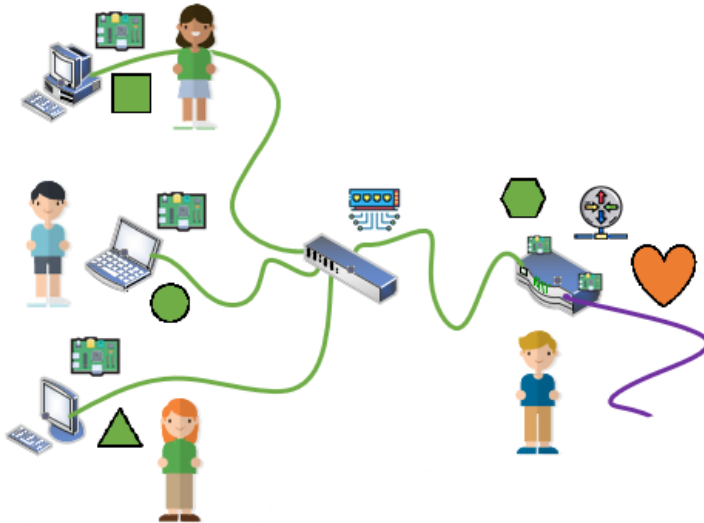


Abbildung 15: Das Netzwerk der Gruppe "Grün" © Knoll, FlatIcon.com, PixaBay.com

Vermittler (Router) besitzen 2 Netzwerkkarten, um in andere Gruppen / Netzwerke (im Beispiel das Orange) weiterzuleiten.

Nachrichtenaustausch in der Gruppe

Nachrichten innerhalb einer Gruppe werden über Symbole, welche Netzwerkgeräte eindeutig identifizieren,¹ gesendet. Dazu wird die Analogie der Post und der Zustellung von Briefen und Paketen zu Hilfe genommen. Der größte Briefumschlag und vorgefertigte Adressetiketten für Absender:innen und Empfänger:innen werden für Nachrichten zwischen den Schüler:innen in einem physikalischen Netzwerk verwendet.



Abbildung 16: Nachrichten zwischen physischen Komponenten werden über deren eindeutige Kennung adressiert, © Knoll.

Symbole werden vom Hersteller der Netzwerkkarte festgelegt. Die Netzwerkkarte am Ziel vergleicht ihr Symbol mit dem auf dem Adressfeld der Nachricht. Ist es gleich, dann ist die Nachricht für sie.

¹ Dies ist eine Abstraktion der so genannten „Media Access Control“ – kurz MAC-Adresse, welche real ein Bitmuster aus 48 Bit ist.

Aufgaben

Der „Zuckerlserver“

Der Zuckerlserver ist ein **Empfänger in einem Netzwerk!** Ihm sendet man eine Anfrage (eine nette Nachricht), in der nach einem Zuckerl gefragt wird und der Server wird eine Antwort in Form von Süßigkeiten geben!



Abbildung 17 <https://freesvg.org/caramel-candy>

Schritte für eine Schülerin:

1. Jede/r bekommt eine **Informationskarte!**
2. **Zeichne** Dein Symbol in den Kreis!
3. Schreibe eine **Nachricht** oder male ein **Bild**.
4. Stecke die **Nachricht** oder das **Bild in den Umschlag**
5. Klebe das **Etikett „NETZWERKKARTE“** auf den Umschlag
6. Male **dein Symbol** in das Feld „Von Symbol“
7. Male das **Symbol der/des Empfängerin/Empfängers** in das Feld „An Symbol“
8. Gib deinen Brief weiter!
9. Die/Der Empfänger/in vergleicht ihr/sein Symbol mit dem Adressfeld „An Symbol“
10. Ein **Empfänger ist der „Zuckerlserver“**- sendet ihm eine nette Nachricht und ihr bekommt Süßigkeiten!

In diesem Abschnitt findet die Kommunikation nur in einem physikalisch verbundenen Netzwerk statt. Der „Zuckerlserver“ ist zum Beispiel das der Router. Zu anderen Gruppen(=Netzwerken) kommt man mit diesen Nachrichten noch nicht!

Erweiterung 1 (Sekundarstufe 1)

Für die Basis reicht diese Analogie und das Versenden von Nachrichten über Symbole. Für die nächste Erweiterung wird eine Überlegung ins Spiel gebracht:

Wie sind wir vorgegangen, um die Nachrichten von einer Person zur anderen zu senden?

Die Antwort ist, dass erfragt wurde, **welche Person welches Symbol** hat. Beispiel: „Susi fragt Peter nach dem Symbol auf seiner Netzwerkkarte“:

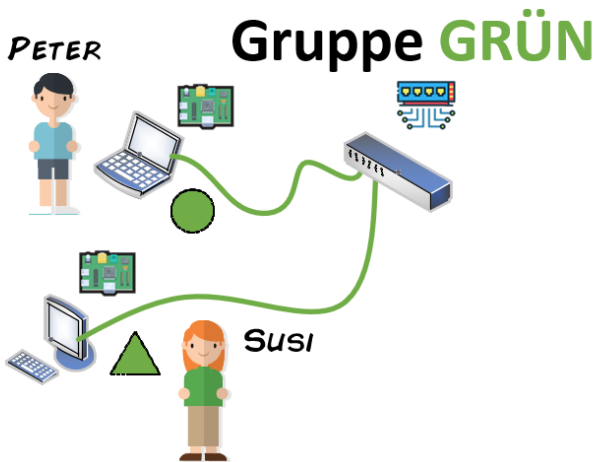


Abbildung 18: Susi und Peter kommunizieren © Knoll, FlatIcon.com, Pi-xaBay.com

Unbewusst wurde hier schon die **Erweiterung in der Adressierung** verwendet. Während **Symbole** vergebene **Kennungen auf Geräten** sind, sind **Namen und Gruppen logischere Kennungen** mit Bedeutung. Im Endausbau des Gesamtnetzwerks kommunizieren am Ende alle Schüler:innen

miteinander, indem die Netzwerke verbunden und logische Adressierung über Namen und Gruppenfarbe¹ eingeführt werden:

Für Nachrichten von Person zu Person gibt es eigene Umschläge und Etiketten:

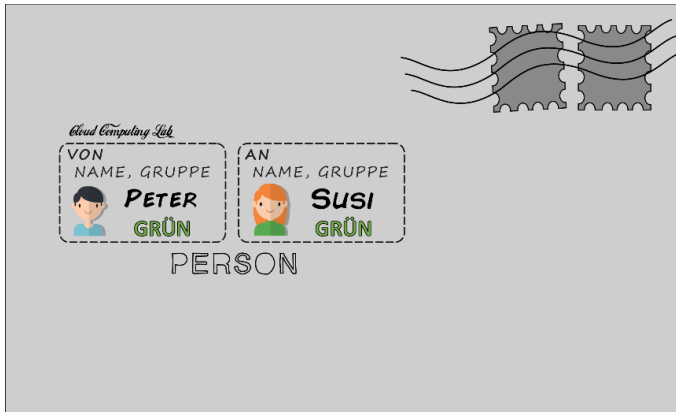


Abbildung 19: Nachrichten zwischen Personen mit logischer Adressierung über Namen und Gruppenfarbe © Knoll

Um diesen Brief zu versenden, muss er als Nachricht in den Umschlag gesteckt werden, der im physikalischen Netzwerk zur Kommunikation dient!

¹ Dies ist die Abstraktion der so genannten „Internet Protocol“ kurz IP-Adresse.

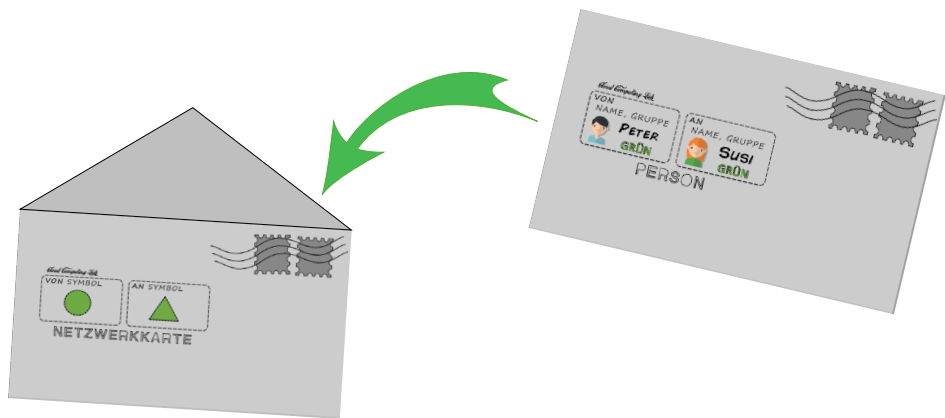


Abbildung 20: Verschachtelung © Knoll

Oft weiß man nicht, wer welches Symbol hat! Das muss man dann innerhalb des eigenen Netzwerks erfragen.

Peter und Susi schreiben sich alle Informationen auf ihre Karten:

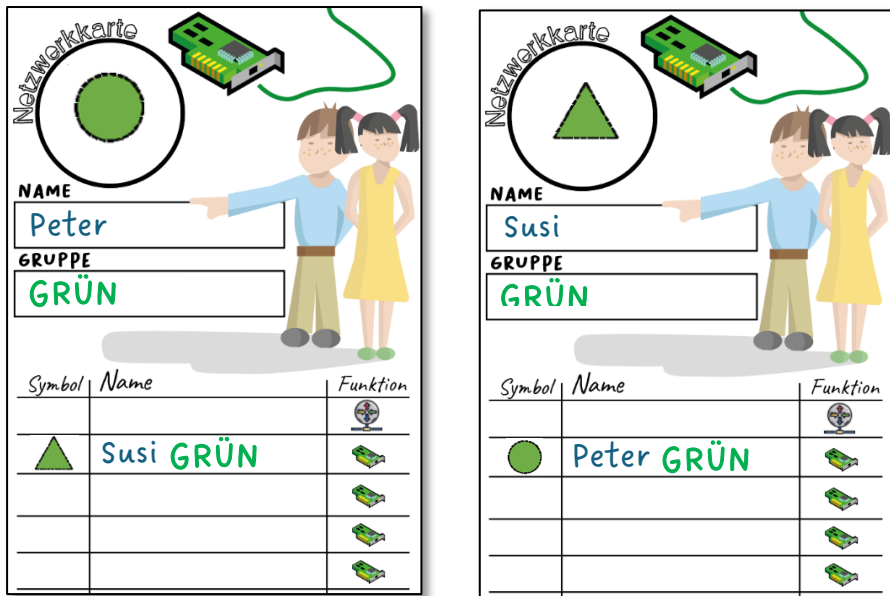


Abbildung 21: Karten von Peter und Susi, auf welchen ihre Informationen vermerkt sind. © Knoll

Damit haben sie die Daten, um beide Umschläge korrekt zu beschriften. Beide haben, da sie im gleichen Netzwerk sind, direkte Verbindungen zueinander und brauchen keine Hilfe, um sich Nachrichten zu senden.

Doch über die Vermittler (Router) sind weitere Netzwerke von anderen Gruppen verfügbar. Symbole von anderen Personen in anderen Netzwerken zu erfragen ist nutzlos, da man sich über einen Vermittler nicht physisch mit anderen Netzwerken verbinden kann:

Wir unterscheiden daher eine ...

- Verbindung im gleichen Netzwerk (direkt) und
- Verbindung über Router (indirekt oder vermittelt.)

Gesamtes Netzwerk aller Gruppen

Jede Gruppe ist in einem Netzwerk. Netzwerke sind über Vermittler verbunden. Jede Netzwerkarte hat eine zugehörige Person!

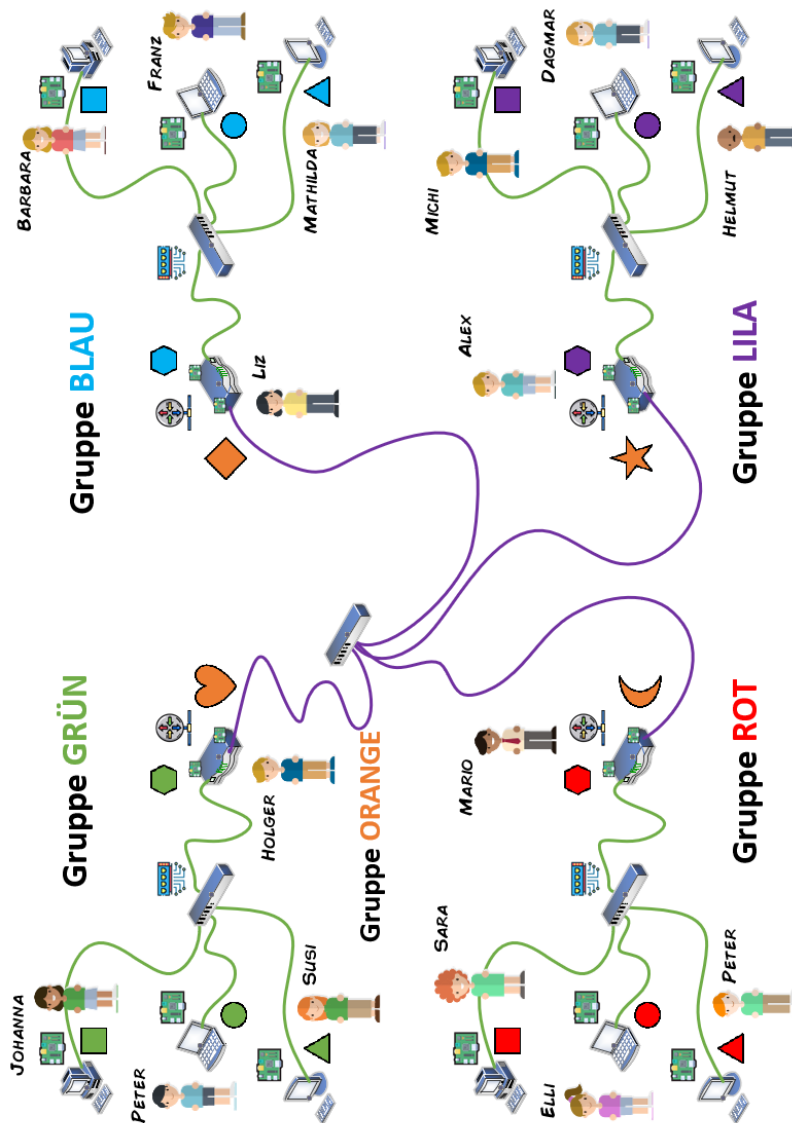


Abbildung 22 Gesamtnetzwerk © Knoll

Nachrichten zwischen Netzwerken

Die Personen kommunizieren mittels ihrer Namen und Gruppen- mittels der „logischen“ Adressierung.

In folgendem Beispiel will Susi eine Nachricht an Barbara senden. Der Brief soll von Gruppe Grün zu Gruppe Blau gehen:

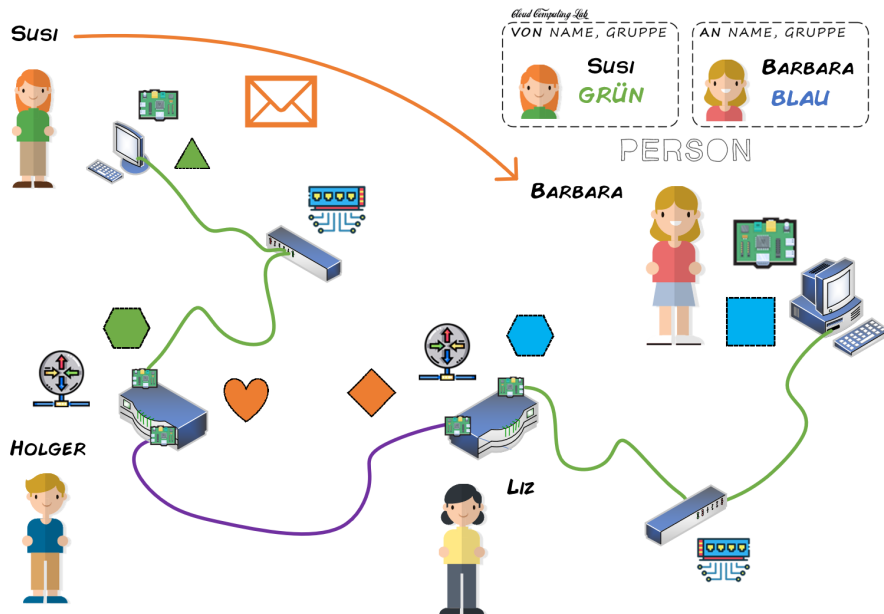


Abbildung 23: Nachricht über Netzwerkgrenzen hinweg © Knoll

Im gleichen Netzwerk finden Computer direkt zueinander und tauschen Daten ohne Umwege aus. Über Netzwerkgrenzen hinweg braucht es einen Vermittler (Router), der die Daten an das korrekte Zielnetzwerk weiterleitet.

In diesem Fall läuft die Kommunikation über die Vermittler Holger und Liz! Susi muss die Nachricht mit den Symbolen an ihren Vermittler (Router) oder auch Gateway namens Holger senden, wenn diese in ein anderes Netzwerk gehen soll.

Beide Vermittler haben besondere Informationen. Sie wissen über welches Symbol in ihrem orangenen Netzwerk sie in welches andere Netzwerk kommen. Diese Informationen können sie auf ihren Karten vermerken:

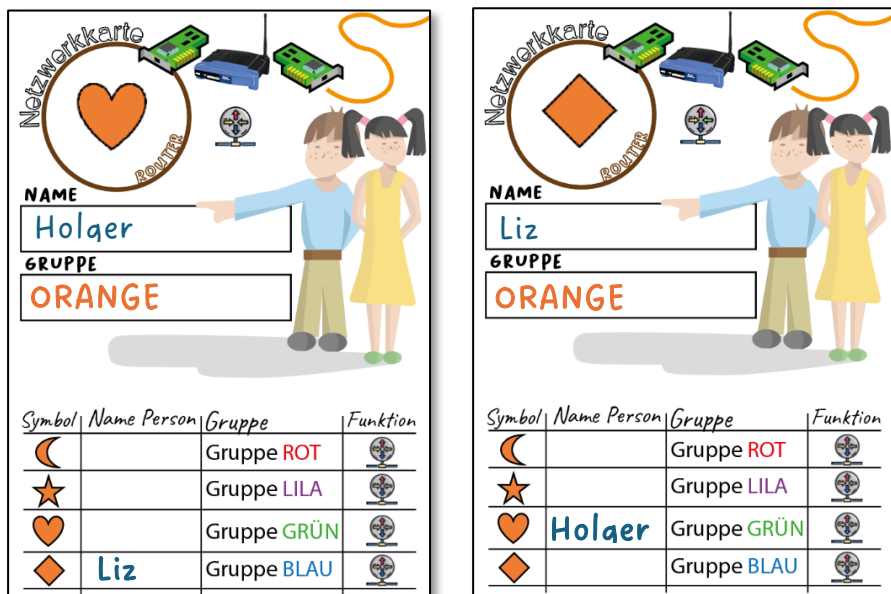


Abbildung 24: Zusatzkarten der Vermittler Holger und Liz © Knoll

Es stellt sich nun die Frage, wie die Briefe auf dem Weg von Susi zu Barbara adressiert werden und wer welche Adresse lesen muss. Die Auflösung ist wie folgt:

Zuerst muss Susi ihre Nachricht mit der logischen Adressierung vorbereiten. Diese geht von ihr, Susi aus der Gruppe Grün zu Barbara aus der Gruppe Blau. Danach folgt die physikalische, direkte Adressierung. Da Barbara außerhalb des Netzwerksegments von Susi ist, muss Susi die Nachricht an ihren Vermittler senden. Sie schreibt an das Symbol von Holger (Vermittler oder Gateway der Gruppe Grün)

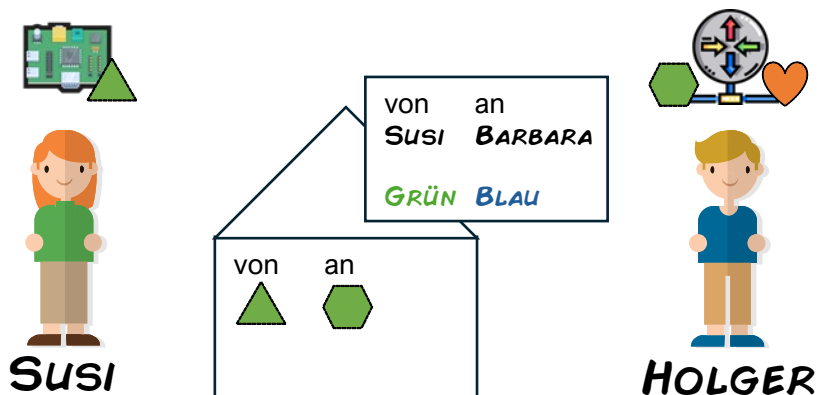


Abbildung 25: Susi sendet eine Nachricht an Barbara © Knoll

Holger bekommt die Nachricht von Susi und öffnet den ersten Brief und sieht sich die logische Adresse des zweiten Briefs an. Für ihn ist wichtig, dass der Brief an die Gruppe Blau geht- er weiß genau an wen er dazu weiterleitet.

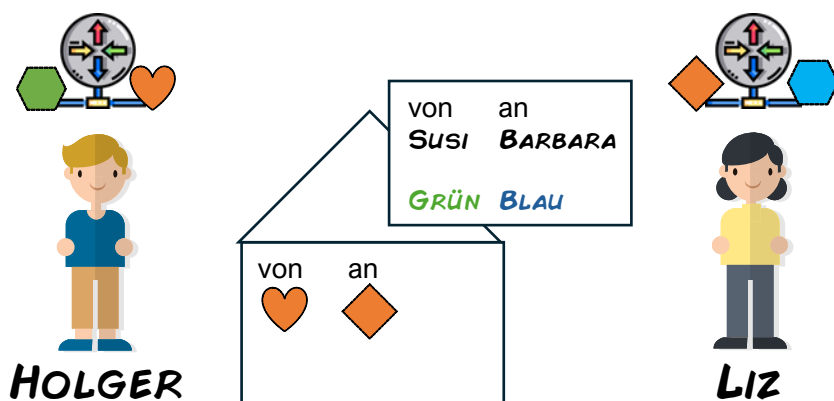


Abbildung 26: Holger leitet an Liz, Vermittlerin von Gruppe Blau, weiter. © Knoll

Liz bekommt die Nachricht nun von Holger und öffnet den ersten Brief ebenfalls. Im Adressfeld des zweiten Briefs steht ihre Gruppe und im Speziellen Barbara als Empfängerin.

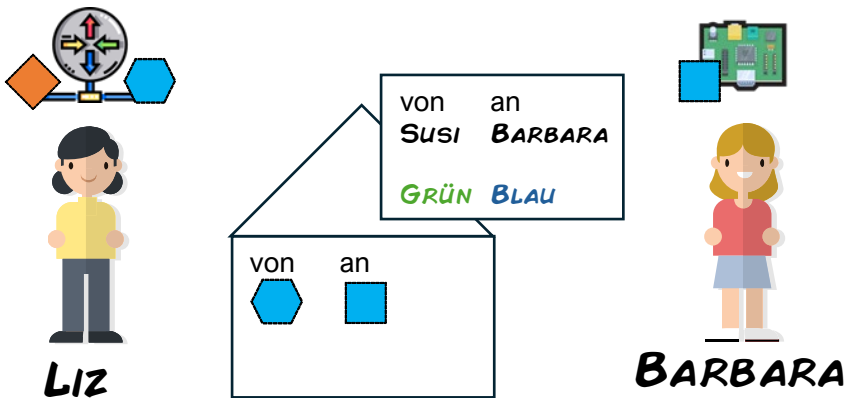


Abbildung 27: Liz leitet die Nachricht in ihrem Netzwerk an Barbara weiter. © Knoll

Aufgaben

Schritte für eine Schüler:in:

1. Vervollständige Deine Karte!
2. Sendet Euch gegenseitig Nachrichten oder Bilder – über Netzwerkgrenzen hinweg!
3. Sendet Anfragen an den „Zuckerlserver“!
4. Beachte die korrekte Adressierung über die Router!

Die Herausforderung ist hier zum einen die Zuordnung von Symbol(=Hardwareadresse) und Namen/Gruppen(=Logische Adresse) zu machen. Zu bemerken ist die Verschachtelung der Nachrichten und die Funktion des Routers, der die logische Adressierung interpretiert und anhand seiner Routinginformationen weiterleitet.

In dieser Erweiterung muss ein Spieler mit einer Netzwerkkarte den „Zuckerlserver“ spielen.

Erweiterung 2 (Sekundarstufe 2)

Mit dieser Erweiterung wird das Konzept von Schichten in der Netzwerk-kommunikation weiter fortgeführt. Die Schüler:innen sind mit den gleichen Regeln und Gegebenheiten in einem Computernetzwerk konfrontiert und müssen Kommunikation unter diesen Voraussetzungen meistern. In diesem Kapitel entwickeln die Schüler:innen ihr eigenes Nachrichtenprotokoll, über welches sie von einem dezidierten „Zuckerl“-Server Süßigkeiten beziehen.

Dienste (Server, Service)

Der „Zuckerlserver“ ist ein Dienst (Server, Service) im Netzwerk. Wenn man ihm eine Anfrage sendet, dann bekommt man eine Antwort in Form von Süßigkeiten! Es gibt jedoch mehrere unterschiedliche Süßigkeiten, die von mehreren Personen (Servern) verwaltet werden!

Um die Anfrage an die richtige Person zu stellen, muss man sich eine für alle gültige **Vorschrift** überlegen (**Protokoll**).

Dafür steht folgendes Adressetikett zur Verfügung:

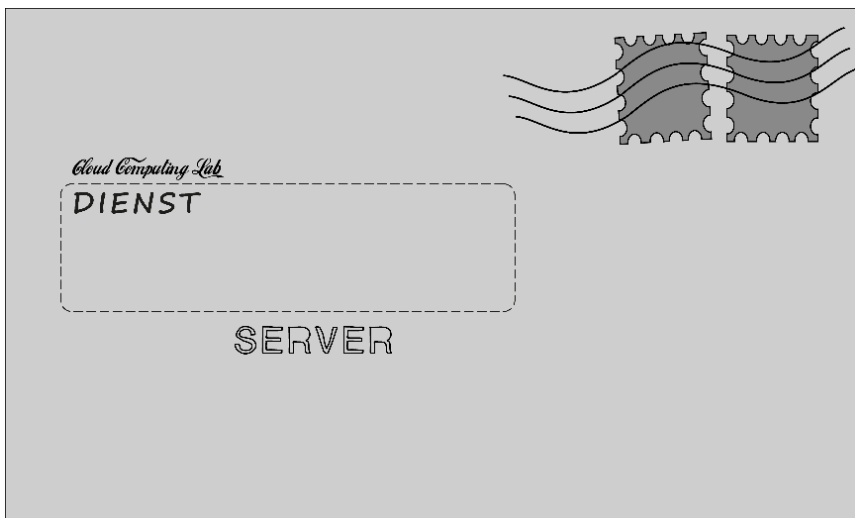
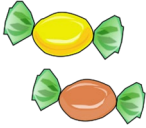


Abbildung 28: Umschlag mit Adressierung an einen Server © Knoll

„Lutschbonbons“ in
Zitrone und Orange



Maximal *3 Stück*
pro Nachricht

Erarbeitetes Protokoll:

Cloud Computing Lab
DIENST

LB

SERVER

1x Zitrone, 2x Orange

„Mini Schokis“ in
Classic, Dunkel und Weiß



Maximal *1 Stück*
pro Nachricht

Cloud Computing Lab
DIENST

MS

SERVER

Weiß

„Schokoriegel“ in
einer Variante



Maximal *1 Stück*
pro Nachricht

Cloud Computing Lab
DIENST

SR

SERVER

Keine Details notwendig

Abbildung 29: Links verschiedene „Zuckerl“-Server und rechts das entsprechende entwickelte Protokoll, © Knoll.

Auf der vorangegangenen Seite wurde eine Vorschrift festgelegt. In dieser bieten 3 Server Süßigkeiten an. Je nach Eigenschaften und Größe muss man dabei genau definieren, welche Angaben erlaubt sind.

Eine konkrete Nachricht nach dem so definierten Kommunikationsprotokoll sieht wie folgt aus:

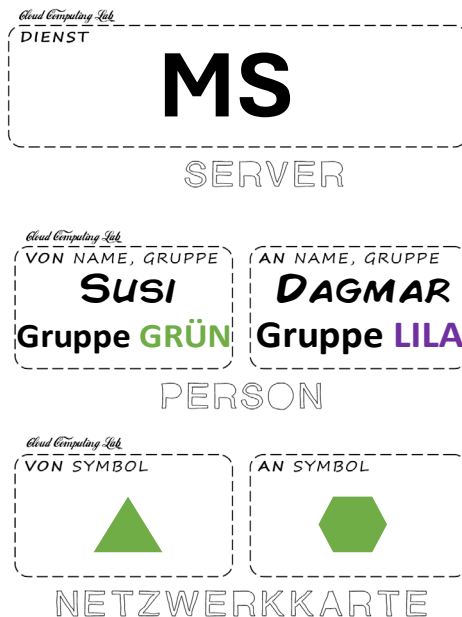


Abbildung 30: Adressfelder aller Briefe, um eine Süßigkeit von einem "Zuckerlserver" zu beziehen. © Knoll

In diesem letzten Schritt wird im Adressfeld des Umschlags ein Server, ein spezieller Dienst adressiert. Im Umschlag werden zusätzliche Daten als Nachricht verpackt, die Details zur Bestellung enthalten.

Aufgabe

Schritte für eine Schüler:in:

1. Überlegt euch eine kurze Kennzeichnung für die 3 Dienste!
2. Welche zusätzlichen Informationen könnte man in die Nachricht schreiben?
3. Zeigt auf einem Flipchart, der Tafel oder über den Beamer, wie euer „Zuckerlprotokoll“ aussieht, damit jede/r Bescheid weiß.

Diese Erweiterung führt an die Funktion eines Transportprotokolls heran. Hier werden Dienste/Anwendungen auf einem Server mit ihrer Kennung angesprochen.

Der/Die Spielleiter/in muss die Entwicklung durch eine Fragerunde steuern:

- Wie könnte man einfach und schnell Lutschbonbons, „Mini Schockis“ oder einen Schockriegel im Adressfeld schreiben. Z.B. LB, MS und SR ...
- Wie könnte die Nachricht im Umschlag aussehen? Z.B.: 1Stk Zitrone, 1Stk Orange

Hier sollte mittels Beamer oder Flipchart, für alle ersichtlich, ein für alle gültiges Protokoll festgehalten werden...

Technischer Kontext

Das Open Systems Interconnect (OSI) Schichtenmodell definiert Schichten im Netzwerkverkehr, die bestimmte Aufgaben erledigen. Es ist eine verfeinerte Version des seit den 1980ern etablierten „Transmission Control“- und „Internet“-Protokoll (TCP/IP) Model. In folgendem Diagramm sieht man den Platz, welchen das Netzwerkspiel in diesen Modellen einnimmt:

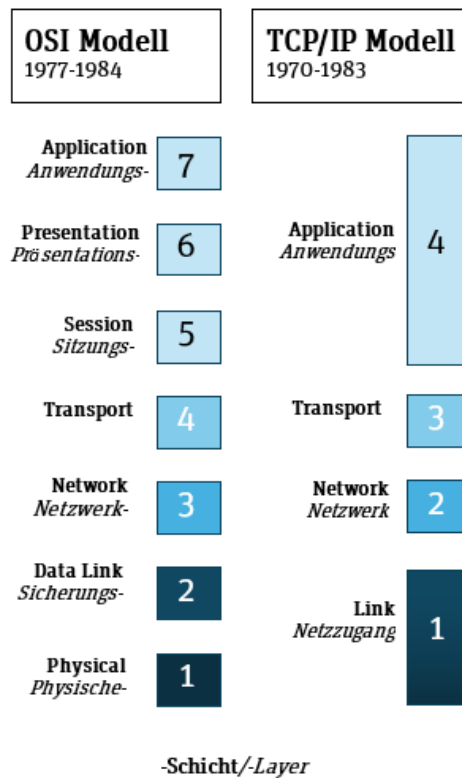


Abbildung 31: Das Netzwerkspiel im realen Kontext (Teil 1), © Knoll.

Netzwerkspiel
In Bezug zu den Modellen

Reale Protokolle
Abstrahiert im Spiel

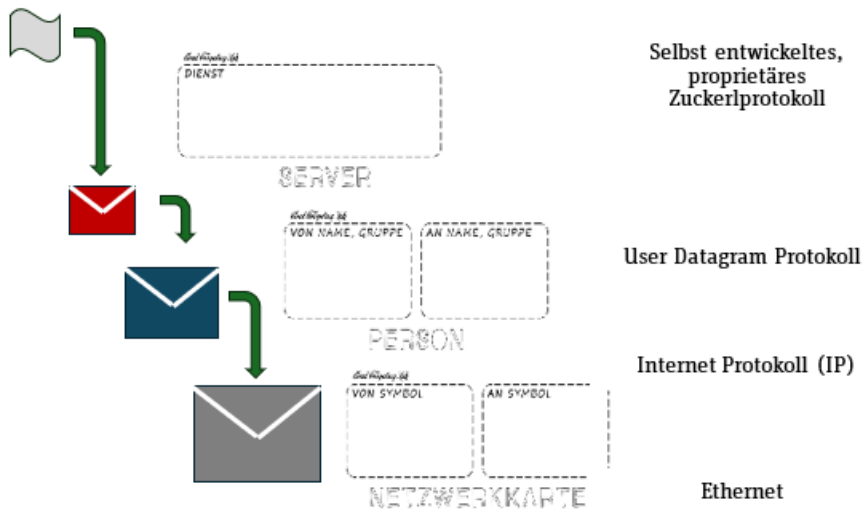


Abbildung 32: Das Netzwerkspiel im realen Kontext (Teil 2), © Knoll.

Zusammenfassung

Mit diesem Spiel werden nicht nur Grundlagen des Cloud-Computings vermittelt, sondern den Schüler:innen auch einen nachhaltigen Zugang zu einem breiten Technologiefeld eröffnet. Durch praxisnahe Einblicke, gemeinsames Lernen und das Arbeiten mit realen Szenarien konnten die Teilnehmenden wertvolle Kompetenzen für Studium, Ausbildung und Beruf entwickeln.

Server

Harald Schwab

Wie funktioniert die Kommunikation im Internet? Wie kommt ein Bild auf einen Bildschirm und woher weiß ein Computer, wo er einen anderen Computer (oft Server genannt) überhaupt findet?

In diesem Workshop lernen Schüler:innen wie Clients (dein Smartphone, dein Tabletts, ...) mit Servern (z.B. einem Webserver) kommunizieren und welche Arten von Servern in einem Netzwerk welche Aufgaben übernehmen. Zwischendurch wird mithilfe eines einfachen Kartenspiels die Funktionalität von Client-Server Kommunikation spielerisch dargestellt.

Dieser Workshop deckt diesen Bereich der Cloud Computing Lab Matrix ab:

Tabelle 2 Cola Matrix: Server © Knoll

		Schulstufen		
		Primärstufe	Sekundarstufe I	Sekundarstufe II
Server	HTTP - Webserver Hypertext Transfer Protocol	X	X	X
	DNS – Namensdienst im Internet Domain Name System		X	X
	DHCP – Hostkonfiguration Dynamic Host Configuration P.		X	X

Was ist ein Server?



Ein Server ist ein besonderer Computer, der spezielle Aufgaben erledigt. Als Beispiel, man möchte in einem Restaurant etwas zum Essen bestellen. Der Kellner bzw. die Kellnerin¹ ist dabei der erste Server, der die Bestellung entgegennimmt. Die Bestellung wird dann an die Küche weitergegeben, was wieder ein anderer Server sein kann. Nachdem die Küche deine Bestellung hergerichtet hat, bringt die Kellnerin bzw. der Kellner das Essen an den Tisch. In der Welt übernehmen Server diese Aufgaben und “servieren” oder “erstellen” für dich Daten oder Informationen.

Wenn man im Internet surft, fragt ein Computer (oder auch Smartphone, Tablet) einen Server nach einer Webseite, die man sich ansehen möchte. Der Server liefert dann die Webseite an das Gerät, damit sie betrachtet werden kann.

Wie im Beispiel mit dem Restaurant erfüllen Server nicht nur eine Aufgabe, es gibt viele unterschiedliche Arten von Servern, die unterschiedlichste Dienste zur Verfügung stellen. Zum Beispiel das Speichern von Dateien oder das Bereitstellen von Spielen im Internet. Sie sind die Helfer, die dafür sorgen, dass Informationen im Internet und in Computersystemen verfügbar sind, wenn wir sie benötigen.

Was macht ein Server

Im Grunde hat ein Server verschiedene Aufgaben, bzw. gibt es für unterschiedlichste Aufgaben eigene Arten von Servern. Im Allgemeinen dient er dazu, Informationen oder Dienste für andere Computer, Geräte und Benutzer zur Verfügung zu stellen. Unter anderem gibt es Server für:

¹ <https://pixabay.com/de/illustrations/kellner-restaurant-bistro-7030934/>

- **Das Bereitstellen und Speichern von Daten:**
Clouddienste wie Dropbox, Google Drive oder OneDrive ermöglichen das Speichern und Teilen von Daten wie Bildern und Dokumenten. Auch ein Webserver stellt Daten bereit – Informationen bzw. Text, der angezeigt wird, wenn eine Webseite aufgerufen wird, ebenso die Bilder und Videos, oder Spiele, die man im Browser spielen kann.
- **Verwalten von Netzwerkressourcen:**
Jeder Computer bzw. Server in einem Netzwerk benötigt zumindest eine eindeutige Adresse – die so genannte IP-Adresse. Diese IP-Adresse ist wie eine Postadresse. Wenn man jemandem einen Brief senden möchte, schreibt man die Adresse auf den Umschlag. Wenn man mit einem Server kommunizieren möchte, brauchst man dessen Adresse. Ein spezieller Server, der solche Adressen vergibt, ist zum Beispiel der DHCP Server¹.
- **Namensauflösung:**
Damit wir uns nicht die IP-Adresse eines Servers merken müssen, gibt es die Möglichkeit diesem zusätzlich einen Namen zu geben. Zum Beispiel www.katzenbilder.at. Für die Zuordnung von diesen Namen zu den Adressen gibt es den DNS-Server². Dieser funktioniert im Grunde wie ein Telefonbuch, wo du nach einem Namen gesucht werden kannst.

Wie kommuniziert man mit einem Server

Um mit einem Server zu kommunizieren, verwendet man in der Regel eine spezielle Software oder Anwendung auf einem Gerät (Computer,

¹ https://de.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

² https://de.wikipedia.org/wiki/Domain_Name_System

Smartphone, ...). Zum Beispiel ein Browser, um eine Webseite von einem Webserver abzurufen. Diese Software, den Browser nennt man Client (Klient). Eine App am Handy kann auch ein Client sein, der zum Beispiel Daten wie das Wetter abfragt. Die Regeln, wie Clients und Server miteinander kommunizieren nennt man Protokoll!

Die grundlegenden Schritte, um mit einem (beliebigen) Server zu kommunizieren sind:

- **Identifiziere den Server:**
Man benötigt die Adresse oder den Namen des Servers, mit dem kommuniziert werden soll. Dies kann eine IP-Adresse (z.B. 192.168.1.2) oder ein Name (z.B. www.katzenbilder.at) sein. Hierbei können wiederum andere Server helfen, wie der DNS-Server, der die Adresse zu einem Namen liefert.
- **Wähle das richtige Protokoll:**
Jeder Dienst, den ein Server bereitstellt, spricht ein bestimmtes Protokoll. Das sind die Regeln, wie zum Beispiel die Nachricht aussehen muss. In der Regel muss man sich selbst nicht darum kümmern, sondern das Programm, das verwendet wird (der Browser als Beispiel) kennt das benötigte Protokoll.
- **Stelle deine Anfrage:**
Mithilfe der Software, die verwendet wird, stellt man dem Server eine Anfrage. Bei einem Webserver wird beispielsweise nach einer bestimmten Seite gefragt. Oder man fragt den E-Mail-Server, ob er eine Nachricht an eine andere Person versenden kann.
- **Warte auf die Antwort:**
Nachdem eine Frage gesendet wurde, wartet man auf die Antwort des Servers. Die Antwort können die gewünschten Daten sein, oder aber auch eine Fehlermeldung, weil

das gewünschte Bild am Webserver nicht vorhanden ist, zum Beispiel.

Arten von Servern im Netzwerk

Es gibt viele unterschiedliche Arten von Servern. Im Folgenden werden drei wichtige Server, mit denen man Tag täglich zu tun hat, auch wenn man es vielleicht nicht merkt, näher vorstellen.

Webserver

Ein Webserver stellt Daten wie Text, Bilder oder Videos bereit, die von einem Client – zum Beispiel einem Browser auf deinem Tablet – abgerufen werden können. Dabei fragt der Client beim Server nach, ob dieser eine bestimmte Ressource – zum Beispiel ein Bild – hat. Wenn der Server dieses Bild hat, schickt der Server dem Client eine OK-Nachricht mit dem gewünschten Bild über das Netzwerk. Wenn der Server dieses Bild nicht kennt, bekommt der Client aber ebenfalls eine Nachricht, jeder bestimmt schon einmal eine solche “404 – Nicht gefunden” Meldung gesehen.

Die Vorschrift, nach welcher Browser und Webserver miteinander sprechen, nennt sich HTTP, das „Hypertext Transfer Protocol“. Übersetzen könnte man das mit „Übertragungsprotokoll für Hypertext“- also für Webseiten.

Unser Spiel funktioniert auf diese Art und Weise. Die Spieler:innen sind die Clients, die nach und nach den Server um eine Karte fragen. Hat der Server diese Karte, bekommt der Client diese. Hat er diese nicht (mehr), bekommt der Spieler stattdessen eine 404 – Karte.

Server für die Zuweisung von Netzwerkadressen

Nicht immer werden Daten von einem Server geholt. Server können auch für die Verwaltung von zum Beispiel Plätzen in einem Computernetzwerk eingesetzt werden. Die Vorschrift für die Kommunikation von

Netzwerkgeräten und diesem Server nennt sich „DHCP“, was für „Dynamic Host Configuration Protocol“ steht. Übersetzt könnte das „Protokoll für die Zuweisung von Netzwerkadressen“ lauten.

In diesem kurzen Spiel wird der Nutzen einer automatischen Platzzuteilung erläutert. Dazu braucht man so viel Stühle wie Spieler:innen:

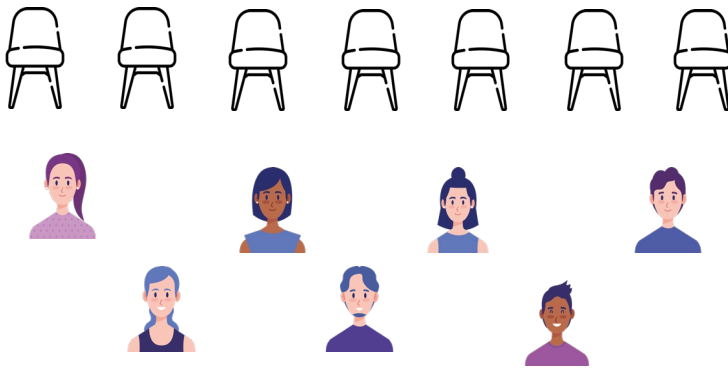


Abbildung 33: DHCP: Stühle und Spieler:innen © Schwab, FlatIcon.com,

Spiele wir vorerst ohne die Hilfe eines Servers, indem sich jede/r Spieler/in geheim die Zahl eines Stuhls wählt:

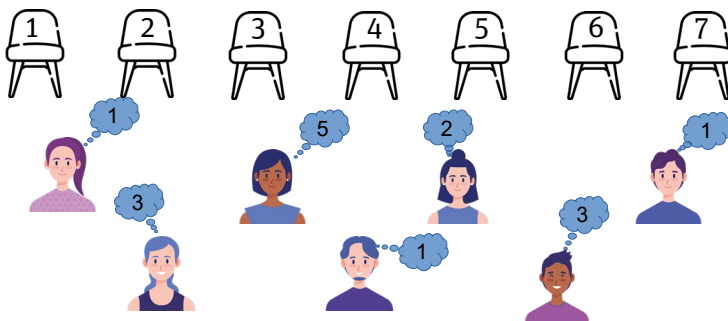


Abbildung 34: DHCP: Stühle nummeriert, Spieler:innen wählen für sich eine Stuhlnummer © Schwab, FlatIcon.com

Im nächsten Schritt gehen die Spieler:innen zu den Stühlen, deren Nummern sie gewählt haben.

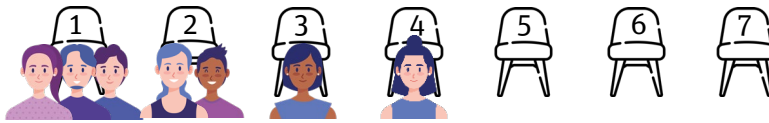


Abbildung 35: DHCP: Auswahl der Stühle ohne ein System © Schwab, FlatIcon.com

Was man sogleich bemerkt ist, dass einige Spieler:innen die gleiche Nummer gewählt haben aber nur eine/r auf dem Stuhl sitzen kann. Sowohl in einem Computernetzwerk als auch in diesem Spiel würde man das einen „Konflikt“ nennen.

Führen wir nun den Sitzplatz-Manager ein:



Abbildung 36: DHCP: Der Sitzplatz Manager, FlatIcon.com

Der Manager hat eine Liste mit allen freien Stuhlnummern. Immer wenn jemand sich hinsetzen möchte, passiert folgendes:

1. Die Schülerin fragt:
„Welchen Sitzplatz soll ich nehmen?“
2. Der Sitzplatz-Manager schaut auf seine Liste und sucht einen freien Platz.

3. Er weist der Schülerin eine Nummer zu, z. B. Stuhl 1.



Abbildung 37 (c) Schwab, FlatIcon.com

4. Die Schülerin setzt sich auf Stuhl 1
– und solange sie im Raum bleibt, „gehört“ dieser Stuhl ihr.

Weil sich nun ein Manager um die Sitzplatzverteilung kümmert, gibt es keine Mehrfachbelegungen mehr. Der Manager weiß genau welche Plätze frei sind:

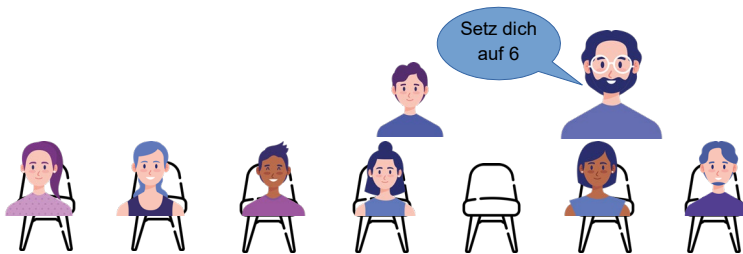


Abbildung 38 DHCP: Kein Chaos mit dem Sitzplatz-Manager © Schwab, FlatIcon.com

Um in einem Netzwerk – zum Beispiel dem Internet – kommunizieren zu können, braucht jede/r Teilnehmer:in eine Adresse. In unserem Fall ist das eine so genannte IP-Adresse, die aus 4 Zahlen (IPv4 : 0 – 255) – zum Beispiel 192.168.1.2 – oder 6 Blöcken aus Hexadezimalzahlen (IPv6 : 0 – 9, A – F) – zum Beispiel 2001:0db8:85a3:0000:0000:8a2e:0370:7334 – bestehen. In einem kleinen Netzwerk, wie zum Beispiel im eigenen zuhause, könnte jedes Gerät händisch eine solche Adresse bekommen. Das wird aber mit mehr und mehr Geräten immer aufwändiger und vor allem unübersichtlich – wer hat den welche Adresse? Es ist vor allem auch unpraktisch, da

bevor man zum Beispiel lossurfen kann, musst erstmal herausgefunden werden, welche Adresse möglich ist und wie man diese auf deinem Gerät einrichtet.

Aus diesem Grund gibt es den DHCP-Server. Dieser vergibt unter anderem jedem Gerät, das sich im Netzwerk meldet, automatisch eine passende IP-Adresse. Zusätzlich gibt der DHCP-Server auch noch weitere Konfigurationen mit, zum Beispiel wo der DNS-Server zu finden ist. Aber er kann auch Aufgaben bzw. Rollen verteilen.

In unserem Spiel vergibt der "DHCP"-Server zwar keine Adressen, aber Aufgaben – beziehungsweise Sonderziele. Wenn man dieses Sonderziel erreicht, bekommt man zusätzliche Punkte. Das erlaubt zum Beispiel 4 Karten mit derselben Zahl als Quartett zu sammeln, oder gibt die doppelte Punktezahl bei vier gelben Karten.

Der Namensdienst im Internet

Während jedes Gerät im Netzwerk eine eindeutige IP-Adresse haben muss, damit untereinander kommuniziert werden kann, ist das speziell für uns Anwender recht mühsam, uns die ganzen IP-Adressen zu merken und zu unterscheiden. Vor allem, sobald IPv6 Adressen verwendet werden. Ähnlich wie bei Telefonnummern. Wie viele unterschiedliche Telefonnummern kennt man auswendig? Die von den Eltern? Vielleicht noch die von den Geschwistern oder der besten Freundin/dem besten Freund? Anstatt sich die ganzen Nummern merken zu müssen, nutzt man ein Telefonbuch. Vielleicht hat man schonmal so ein altes dickes gelbes Telefonbuch in der Hand gehalten? Zumindest auf dem Smartphone hat man von allen Leuten, die man kennst, die Nummer eingespeichert.

Ähnlich funktioniert das im Netzwerk bzw. Internet. Anstelle uns die ganzen IP-Adressen merken zu müssen, geben wir einem Computer oder Server einfach einen Namen. Der englische Begriff für diesen Namen ist „Domain“ (Domäne) und heißt eigentlich „Bereich“, „Zuständigkeitsbereich“ oder „Gebiet“. Eine Domain ist also ein eigener Bereich im

Internet, den jemand besitzt oder verwaltet. Diese besteht zumindest aus zwei, oft auch aus drei Teilen. „example.com“ oder „www.katzenbilder.at“ zum Beispiel. Welche (IP)-Adresse sich hinter dem Namen verbirgt, das weiß der DNS-Server. Wenn man – oder besser gesagt der Browser – wissen will, wie die eigentliche Adresse von www.katzenbilder.at lautet, fragt er einfach den DNS-Server. DNS steht für das „Domain Name System“ - also ein System für Namen einer Domäne.

Material

Für dieses Spiel wurden Karten des beliebten Spiels UNO verwendet. Ein paar Karten wurden im Nachhinein mit ablösbarer Folie beklebt.

Ein übliches UNO-Kartenset besteht aus den Farben Rot, Grün, Blau Gelb und enthält jeweils 2 Karten pro Zahl und Farbe (Zahlen 1 – 9) und jeweils eine Farbe mit der Zahl 0.

Je nach Gruppengröße sollte der Spielleiter/Server alle Zahlenkarten auf der Hand haben (5-7 Spieler:innen) oder aus einer Hälfte der Zahlenkarten (2-4 Spieler:innen) – Zahlen 0 – 4 zum Beispiel¹. Zum einen erleichtert das den Kartenstapel in der Hand zu halten, zum anderen werden dadurch die einzelnen Runden nicht zu lange, wodurch mehr Runden gespielt werden können. Auch ermöglicht das, mit einem Set Karten mehrere Gruppen zu bilden.

Es sollten zumindest mehr als 4 Zahlen zur Verfügung stehen, damit nicht sofort ein Dead-Lock², wodurch am Ende kein einziger Spieler:in ein Quartett zusammenbekommt.

¹ Damit ist es möglich ein UNO-Kartenset auf zwei kleinere Gruppen aufzuteilen, indem für jede Gruppe die Hälfte der Zahlen zur Verfügung steht.

² Mit Dead-Lock ist in diesem Spiel gemeint, dass 2 oder mehr Spieler:innen auf dieselbe Farbe gehen – entweder, weil sie diese bewusst sammeln wollen, oder aus taktischen Gründen, um einem/r Gegenspieler:in die Chance auf ein Quartett zunichtezumachen.

Beispiel: Gruppe 1 wird von Lehrer:in A geführt. Diese nimmt die Karten 0 – 4 und die Hälfte der zur Verfügung stehenden Farbwahl/+4 Karten. Gruppe 2 wird von Lehrer:in B geführt und nimmt die restlichen Karten. Zu Spielbeginn wird den Schüler:innen kurz bekannt gegeben, welche Karten möglich sind zu sammeln, damit nicht direkt 404-Karten ausgegeben werden.

Die Sonderkarten (Runde aussetzen, +2, Richtungswechsel) werden in diesem Modus nicht verwendet.

Basisspiel (Primarstufe)

Rollen

In diesem Spiel gibt es zwei Rollen. Alle Mitspieler:innen sind die Clients. Eine Person übernimmt die Rolle des Webservers – der oder die Spielleiter:in. Der Webserver hat den Stapel mit den Karten über und gibt diese an die Clients – die Spieler:innen – aus. Dabei wird das Client-Server Prinzip genutzt. Abwechselnd (Reihum) fragen die Clients den Server nach einer Karte. Hat der Server die gewünschte Karte noch auf der Hand, bekommt der Client diese Karte. Hat der Server die Karte nicht oder nicht mehr, dann bekommt der Client eine 404 Karte.

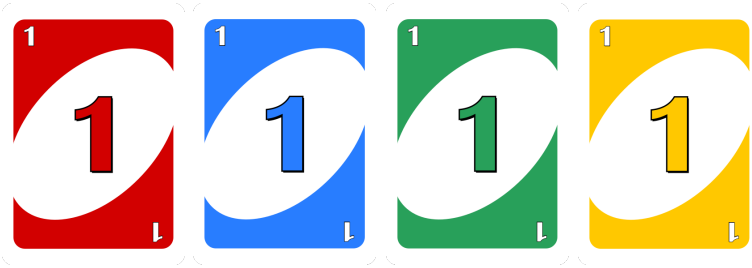
Die Anfrage passiert dabei laut/öffentlich:



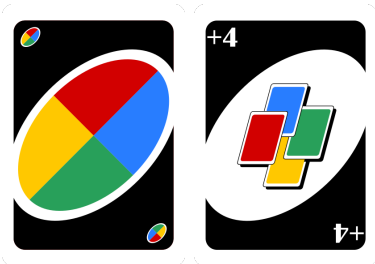
Die Karte wird verdeckt vom Server an den Client gegeben:



Ziel ist es, ein bzw. mehrere vollständige Quartette/Vierlinge zu sammeln. Ein solches Quartett bzw. Vierling sind 4 Karten derselben Zahl in unterschiedlichen Farben, zum Beispiel:



Fragt man um eine Karte, die es nicht (mehr) auf der Hand des Servers gibt, erhält man eine 404 Karte, repräsentiert durch die Farbwahl – bzw. +4 Karte aus dem UNO-Spiel.



Das Spiel endet spätestens, wenn der Spielleiter/Server keine 404 bzw. Zahlenkarten mehr auf der Hand hat.

Am Ende werden die Punkte zusammengezählt – für jedes vollständige Quartett/Vierling erhält man einen Punkt, für jede 404 Karte wird 1

Punkt abgezogen. Der/die Spieler:in mit den meisten Punkten gewinnt – bei Gleichstand gibt es mehrere Sieger. Beziehungsweise werden damit Rundenpunkte gesammelt und nach mehreren Runden ergibt sich so ein/e Sieger:in.

Beispielrunde

Im Folgenden wird eine beispielhafte Runde skizziert:

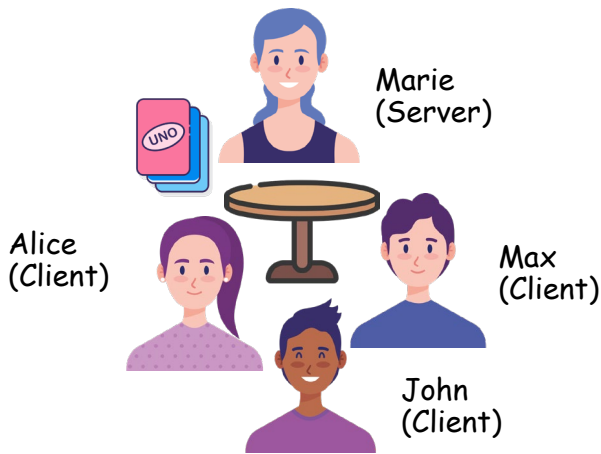


Abbildung 39 Beispielrunde © Schwab, FlatIcon.com

Die Runde beginnt mit einer einfachen Anfrage:

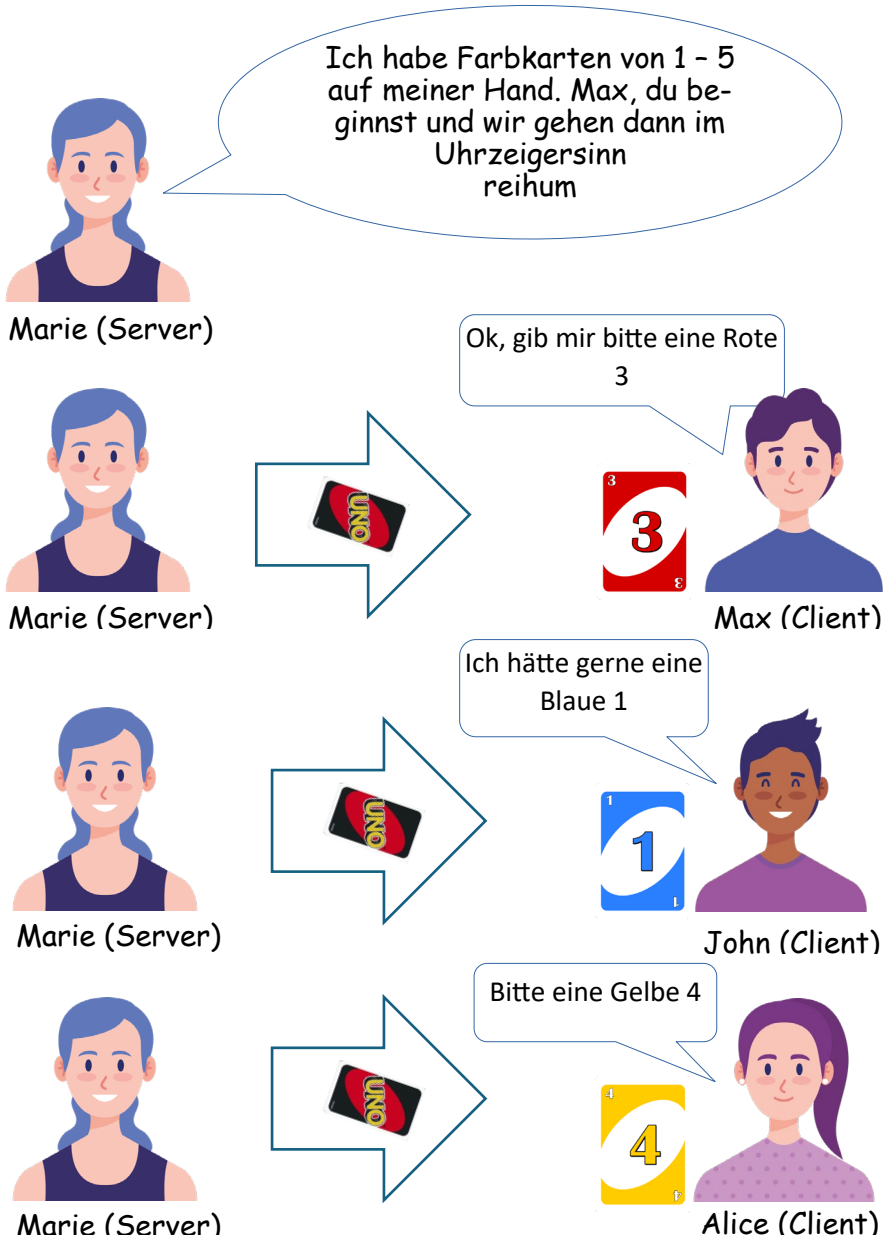


Abbildung 40 Spielverlauf: Anfrage © Schwab, FlatIcon.com

Am Ende werden die Punkte zusammengezählt. Pro Karte oder Quartett bekommt ein Client/Spieler:in einen Punkt. Pro 404 Karte wird ein Punkt abgezogen. Der/Die Spieler:in mit den meisten Punkten gewinnt. Bei Lust und Laune können auch mehrere Runden gespielt werden.

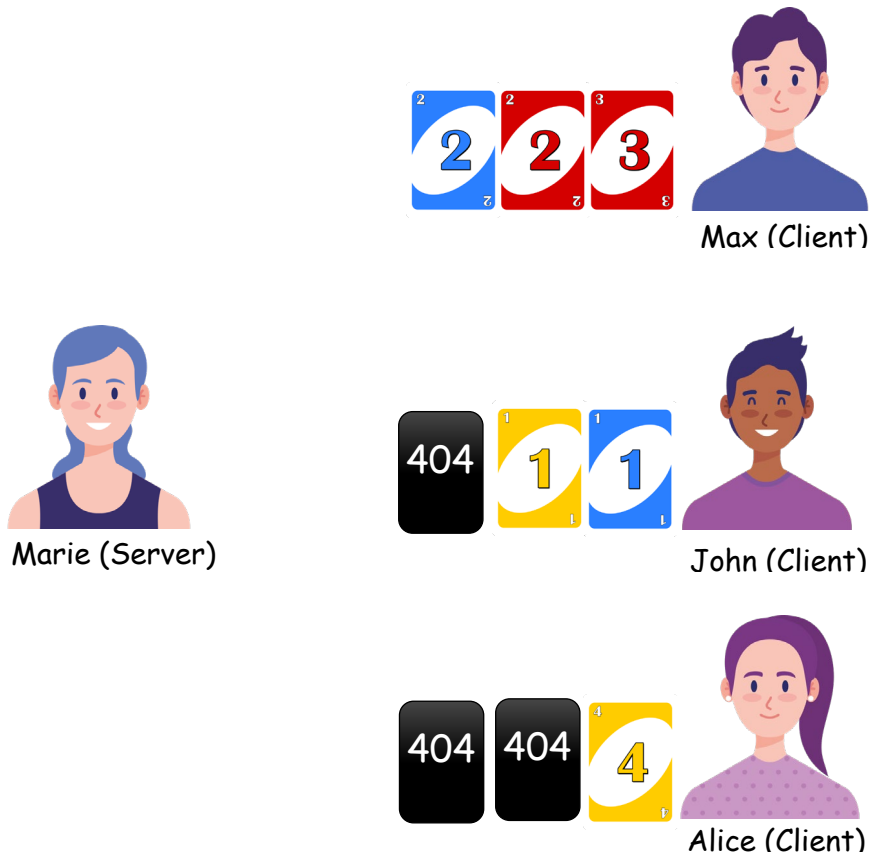


Abbildung 41 Spielverlauf: 404 Not Found © Schwab, FlatIcon.com

Erweiterung 1 (Sekundarstufe 1)

Im Web kann man nicht nur Daten abfragen, sondern man kann auch Daten senden. Dabei spricht man von einem GET-Request (Abfrage) bzw. POST-Request (Senden)

Im Basisspiel hatten wir nur Anfragen mit GET genutzt. Die Spieler:innen fragten Daten vom Server ab- bekamen Karten. In dieser Erweiterung gibt es die Möglichkeit für Clients, Daten an den Server (zurück) zu senden: Sobald ein/e Client/Spieler:in eine 404-Karte erhalten hat, kann diese in einer folgenden Runde wieder zurückgegeben werden. Dabei muss die 404-Karte zusammen mit einer Zahlenkarte verdeckt an den Server gegeben werden

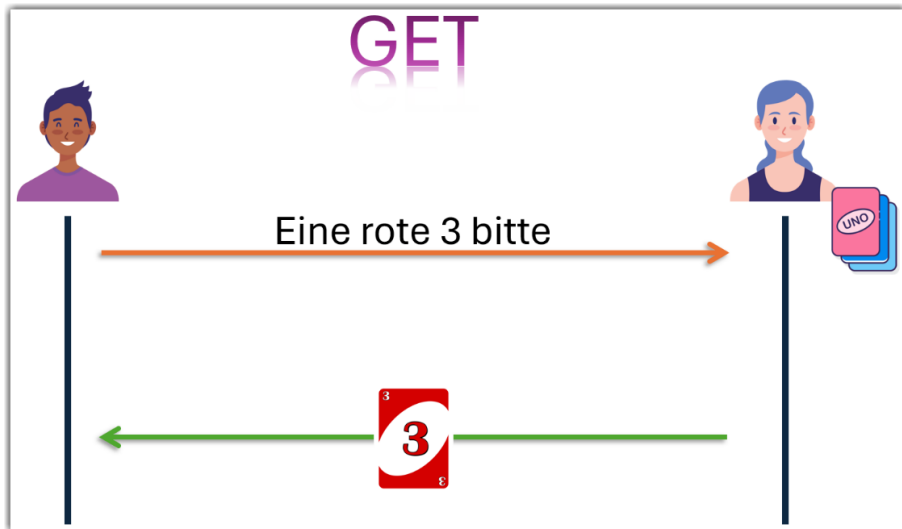


Abbildung 42: Anfrage mit GET

Damit landen Zahlenkarten wieder zurück, die anderen Mitspieler:innen wissen aber nicht, welche. Die Dynamik und das taktieren wird dadurch komplexer, was das Spiel aber auch spannender macht.

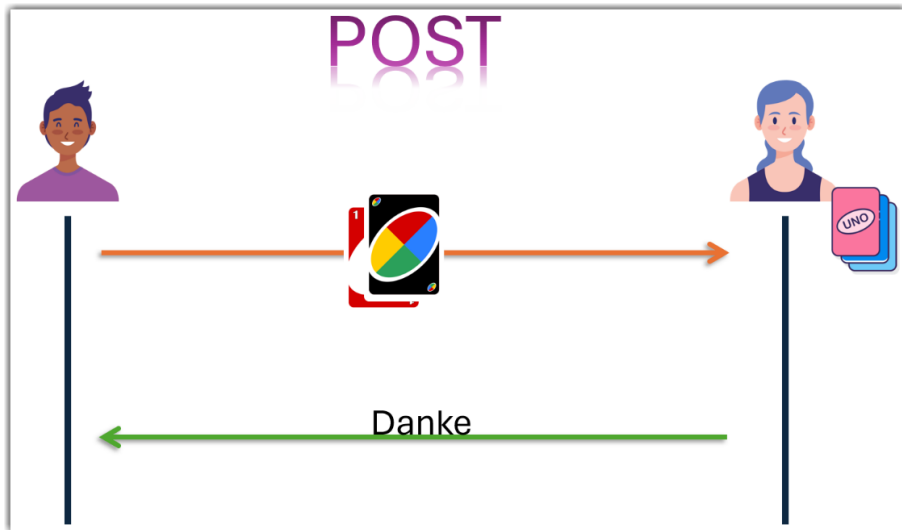


Abbildung 43: Anfrage mit POST

Nutzt ein/e Spieler:in diesen POST-Request, kann in dieser Runde zwar keine neue Karte erhalten werden, es wird aber zumindest eine 404-Karte – die am Ende Punkteabzug bedeuten würde – abgelegt.

In der Realität erhält ein Client bei einem POST-Request auch immer eine Antwort vom Server. Dies ist oft aber einfach nur eine Bestätigung, dass die übertragenen Daten korrekt verarbeitet wurden – oder eben auch nicht. Für unser Spiel haben wir diesen Teil etwas vereinfacht und nutzen POST einfach nur dazu, Karten wieder abzulegen.

Erweiterung 2 (Sekundarstufe 2)

Wie in der Einleitung bereits erläutert, gibt es unterschiedlichste Server in einem Netzwerk bzw. der Cloud. Neben Webserver, die uns Webseiten und Services zur Verfügung stellen, spielen auch DNS und DHCP-Server im Internet eine wichtige Rolle. Während ein DNS-Server als eine Art Telefonbuch gesehen werden kann, der zu den Adressen (IP-Adressen) die passenden Namen (Domain) kennt, gibt ein DHCP-Server die entsprechenden IP-Adressen aus. Das ist besonders im lokalen Netzwerk – aber auch in der Cloud selbst – eine wichtige Rolle, von der wir als Anwender so gut wie nichts mitbekommen.

Ein DHCP-Server gibt dabei nicht einfach nur IP-Adressen aus. Mittels DHCP können auch Konfigurationen verteilt werden. In unserem Spiel wird der DHCP Server Sonderziele an die Clients, die Spieler:innen vergeben.

Der DHCP-Server ist dabei ein Stapel Karten. In diesem Stapel befinden sich sowohl Zahlenkarten (Rot 1, Blau 3, etc.), sowie auch Sonderkarten (z.B. +2).

Zum Beginn zieht jede/r Spieler:in eine Karte von diesem Stapel. Je nach Karte ergibt sich nun für jede/n Spieler:in ein Sonderziel:

- Zieht ein/e Spieler:in eine Zahlenkarte, ergibt ein Quartett/Vierling dieser Zahl die doppelten Punkte
- Zieht ein/e Spieler:in eine Sonderkarte (zB +2 Rot) gelten als Quartett/Vierling 4 Karten dieser Farbe

Die Sonderziel-Karte wird nicht auf der Hand des/der Spieler:in gehalten, sondern neben verdeckt abgelegt. Sobald das Sonderziel erreicht ist, werden die dazu gesammelten Karten ebenfalls verdeckt abgelegt und eine weitere Sonderziel-Karte vom Stapel (DHCP-Server) gezogen. Am Ende werden wieder alle Punkte zusammengezählt und die 404-Karten abgezogen. Der/Die Spieler:in mit der höchsten Punktezahl gewinnt. Ein POST wie in Erweiterung 1 ist natürlich auch möglich.

Speicher

Michael Brickmann, Mathias Knoll

Lerne, wie Computer Informationen speichern und verarbeiten. Welche Informationen werden wie repräsentiert? Unser Workshop zeigt Dir aus welchen Bestandteilen Computer bestehen und wo und wie Daten gespeichert werden.

Dieser Workshop deckt diesen Bereich der Cloud Computing Lab Matrix ab:

Tabelle 3 Cloud Matrix: Speicher © Knoll

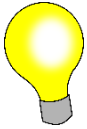
Speicher	Grundlagen der Datenspeicherung Mechanisch, Flip-Flop Schaltung, ...	X	X	X
	Datenschutz und -sicherheit		X	X

Im Verlauf dieses Workshops wurde Anschauungsmaterial und das Spiel „Turing Tumble“ verwendet, um einfache Schaltkreise in einem Rechner darzustellen. In weiterer Folge kam auch das Spiel „Spintronics“ für den gleichen Zweck zum Einsatz.¹ Letzteres vermittelt eine mechanische Version der Elektrotechnik und bietet eine genauere Sichtweise auf Schaltkreise und Logikgatter.

Ein Vortrag über Datenschutz- und -sicherheit fördert das Bewusstsein über die Datenablage in der „Cloud“ und rundet das Programm dieses Workshops ab.

¹ Siehe <https://upperstory.com/de>

Bits (binary digits) & Bitfolgen



Ein Bit ist wie ein kleiner elektronischer Lichtschalter. Es kann entweder ein oder aus sein, also entweder an oder ausgeschaltet.

Wie eine Glühbirne¹: Wenn sie leuchtet, ist der Schalter an, und wenn sie nicht leuchtet, ist der Schalter aus. In der Computerwelt repräsentiert ein Bit genau diese zwei Zustände: an oder aus, 1 oder 0.

Mehrere Bits beinhalten komplexere Informationen. Zum Beispiel: Mit 2 Bits kann man 4 verschiedene Kombinationen von 0en und 1en machen- und zum Beispiel für einen Kompass verwenden:

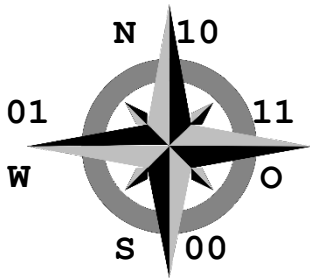


Abbildung 44 Kompass , Pixabay.com

Bits sind die kleinsten Bausteine der digitalen Welt und werden verwendet, um alle Arten von Informationen in Computern zu speichern und zu verarbeiten, von Texten und Bildern bis hin zu Videospielen und Internetseiten. Sie sind die Grundlage dafür, wie Computer arbeiten und Informationen verarbeiten können.

¹ <https://pixabay.com/vectors/bulb-electric-idea-light-yellow-147785/>

Elementarer Bestandteil im Computer sind Schalter, die mechanisch, elektronisch mit Röhren oder Transistoren umgesetzt wurden. Heute finden wir Milliarden Transistoren ("Transfer Resistor") auf kleinster Fläche. Funktionsweise anhand eines mechanischen Modells:

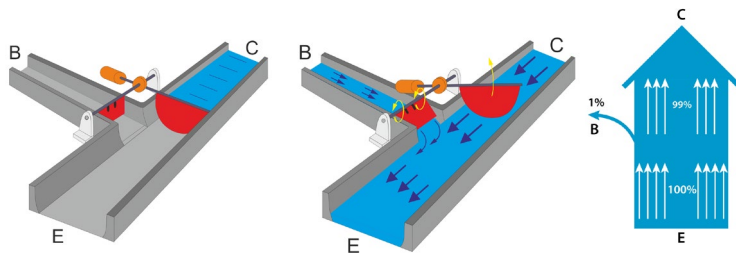


Abbildung 45 Analogie, wie Transistoren funktionieren¹

Mit Transistoren können einfachste logische Schaltungen umgesetzt werden. Diese nennen sich logische Gatter. Logische Werte sind die Wahrheitswerte „Wahr“ (true, abgekürzt T) und „Falsch“ (false, abgekürzt F). Auf den beiden Wahrheitswerten sind die Booleschen Operationen „not“, „and“, „or“, und „xor“ („exklusives oder“) definiert:

not		and	F	T	or	F	T	xor	F	T
F	T	F	F	F	F	F	T	F	F	T
T	F	T	F	T	T	T	T	T	T	F

Abbildung 46 Logische Operationen: $F(\text{alse})$ und $T(\text{rue})$

¹ <https://www.leifiphysik.de/elektronik/transistor/grundwissen/der-transistor-effekt> [05.11.2025]

Ein AND und ein OR Gatter können mit jeweils 2 Transistoren umgesetzt werden:



Abbildung 47 AND-Gatter: Nur beide Knöpfe ergeben Licht (Wahr) Bilder © Knoll

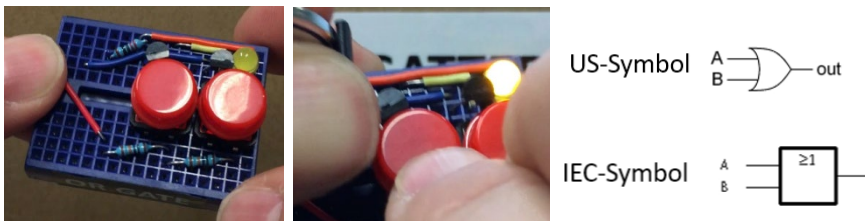


Abbildung 48 OR-Gatter: Jeder einzelne oder alle Knöpfe ergeben Licht (Wahr) Bilder © Knoll

Flip-Flop Speicherung eines Bits

Ein Flip-Flop wirkt wie ein Schalter: Die Grundstellung ist mit 0 an x und y. Das temporäre Anlegen von 1 an x setzt den Output auf 1. Das temporäre Anlegen von 1 an y setzt Output auf 0. Die Schaltung ist damit geeignet zur Speicherung einzelner Bits

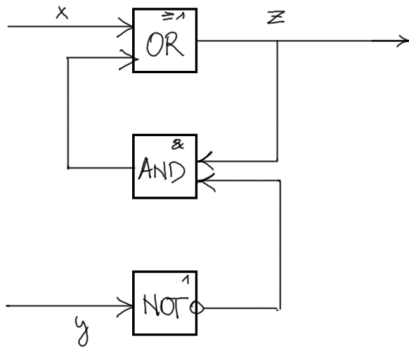


Abbildung 49 Flip-Flop Schaltung © Knoll

Legt man an die Schaltung am Schalter X eine Spannung an, resultiert das in einer Spannung an Z. Auch wenn die Spannung auf 0 geht, bleibt das Bit gesetzt:

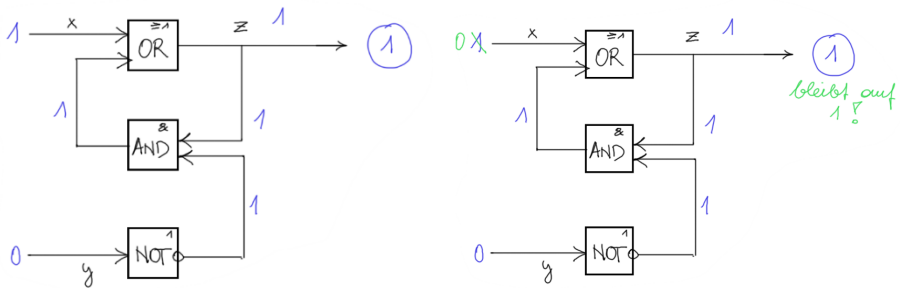


Abbildung 50 Flip-Flop - 2 Diode einschalten © Knoll

Erst wenn eine Spannung an Y angelegt wird, schaltet Z auf 0. So bleibt es auch, wenn die Spannung auf Y wieder 0 wird:

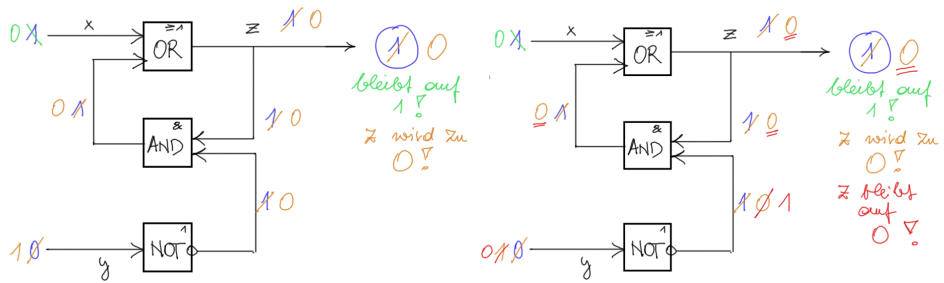


Abbildung 51 Flip Flop: Diode ausschalten © Knoll

Das Spiel „Turing Tumble“ ist ein mechanischer Computerbaukasten, der logische Abläufe über Kugeln, Schwerkraft und Bauteile wie Rampen, Kreuzungen und insbesondere Schaltern sichtbar macht. Sein zentraler pädagogischer Nutzen liegt darin, abstrakte digitale Konzepte – etwa den Zustandsspeicher eines Schalters – physisch erfahrbar zu machen.

Beim Durchlauf der Kugeln kippen die Schalter abwechselnd nach links oder rechts und repräsentieren damit klar erkennbar die binären Zustände 0 und 1. Lernende sehen unmittelbar, wie ein Flip-Flop seinen Zustand speichert und wie dieser Zustand nachfolgenden Operationen beeinflusst.

Damit bietet Turing Tumble eine intuitive Brücke zwischen mechanischer Aktion und digitaler Logik, ideal für den Einstieg in Informatik- und Elektrotechnikgrundlagen. Logische Schaltungen kann man sehr gut mit dem Spiel „Turing Tumble“ kennenlernen und ausprobieren.

Turing Tumble eignet sich hervorragend, um das Funktionsprinzip eines Flip-Flop-Speichers anschaulich zu verstehen. Die mechanischen Schalter im Spiel arbeiten wie ihre elektronischen Pendanten: Jede herabfallende Kugel kippt den Mechanismus von einer stabilen Stellung in die andere. Dadurch entsteht ein klar sichtbarer bistabiler Zustand, der entweder „links“ oder „rechts“ ist – ein direktes Analogon zu 0 und 1 in der digitalen Elektronik.

Durch die physische Bewegung der Bauteile wird erfahrbar, wie ein Flip-Flop Informationen speichert und wie dieser gespeicherte Zustand nachfolgende Logikpfade beeinflusst. So vermittelt Turing Tumble die abstrakte Idee des digitalen Speichers auf intuitive und spielerische Weise:

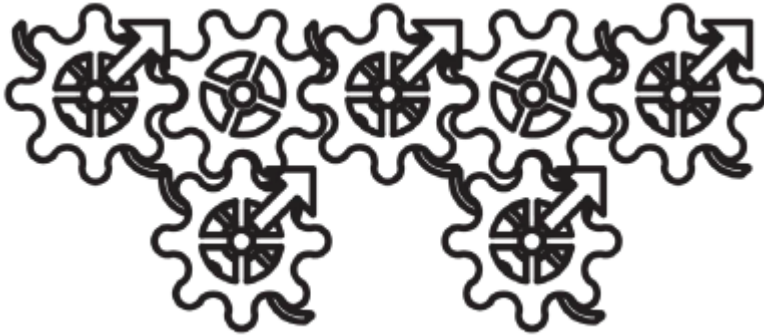


Abbildung 52 Flip Flop in Turing Tumble (UpperStory, 2024)

Die drei oberen Zahnräder sind kleine Schalter, die sich merken können, in welche Richtung sie zeigen.

- Lässt man eine Kugel auf das mittlere obere Zahnrad fallen, prüft die Kugel nur, in welche Richtung das Zahnrad zeigt. Der Zustand wird nicht verändert – man testet also nur nach, ohne umzuschalten.
- Lässt man eine Kugel auf das linke obere Zahnrad fallen, wird das Zahnrad immer auf „links“ gestellt, egal wie es vorher stand.
- Lässt man eine Kugel auf das rechte obere Zahnrad fallen, wird das Zahnrad immer auf „rechts“ gestellt, egal wie es vorher stand.

So ein Zahnrad-Schalter verhält sich wie ein Flip-Flop in einem Computer. Ein Flip-Flop ist ein winziges Bauteil, das ein einzelnes Bit speichern kann – also eine Information, die entweder 0 oder 1 ist. Ein Computerspeicher besteht aus Milliarden solcher Flip-Flops.

Datenbanken

Priska Steininger

In diesem Workshop lernen Schüler:innen auf spielerische Art, wie Daten modelliert und in einer Datenbank gespeichert sowie abgerufen werden können.

Die Baustein-Datenbank soll zeigen, warum die Datenmodellierung wichtig ist und welche Bausteine dazu benötigt werden. Dazu gehören die Entitäten (Dinge), also zum Beispiel der Baustein, dessen Eigenschaften (wie z. B. Farbe oder Form) sowie die Verbindungen zu anderen Bausteinen oder zu Figuren. Die Datenmodellierung beschreibt also „WAS“ in einer Baustein-Datenbank (Bausteinsammlung) zu speichern bzw. aufzubewahren ist und wie die Dinge zueinander in Verbindung stehen.

Dieser Workshop deckt diesen Bereich der Cloud Computing Lab Matrix ab:

Tabelle 4 Cloud Matrix: Datenbanken © Knoll

Datenbanken	Strukturierung von Information (Entity-Relationship Modell)	X	X	X
	Datenbankabfragen (Structured Query Language)		X	X
	Fortführende Konzepte (BigData und noSQL)			X

Zunächst soll der Begriff „Datenbank“ spezifiziert werden. Eine Datenbank ist ein Ort, wo man auf organisierte Weise Daten ablegen bzw. speichern können. Grundsätzlich gibt es viele Datenbanken, die man hierfür verwenden kann. Die zwei wesentlichen Datenbank-Kategorien sind aber SQL und NoSQL (Zhao, 2022).

SQL bedeutet Structured Query Language, also strukturierte Abfragesprache. SQL-Datenbanken werden auch als relationale Datenbanken bezeichnet, da sie aus Relationen/Tabellen bestehen. Dies bedeutet, dass viele Tabellen, die miteinander verbunden sind, eine Datenbank bilden. An dieser Stelle ist es wichtig zu erwähnen, dass SQL-Datenbanken stets ein vordefiniertes Schema erfordern. Bevor Sie also eine Tabelle anlegen, müssen Sie die Struktur entsprechend festlegen wie etwa Spalten der Tabelle und die jeweiligen Datentypen wie z. B. Ganzzahl oder Zeichenketten etc. vgl. (Zhao, 2022)

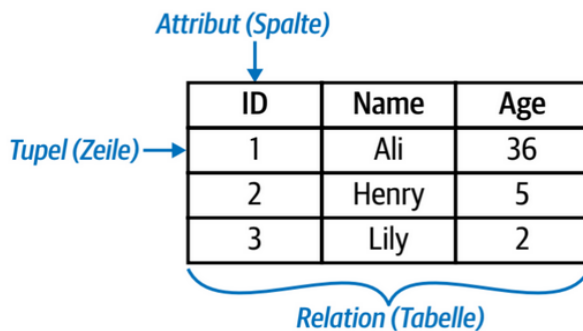


Abbildung 53 Tabelle (Zhao, 2022)

Lassen sich Daten nicht mehr in einer strukturierten Weise organisieren, benötigen Sie eine NoSQL-Datenbank. Dies bedeutet Not only SQL, also nicht nur SQL. Eines der wesentlichen Merkmale von NoSQL ist, dass sie dynamische Schemata besitzen, was bedeutet, dass ihre Schemata nicht im Vorhinein festgelegt werden müssen. Die Daten liegen also nicht mehr in einer strukturierten Tabelle vor. Die nachfolgende Abbildung zeigt, wie die Daten in einer NoSQL-Datenbank wie MongoDB gespeichert werden vgl. (Zhao, 2022).

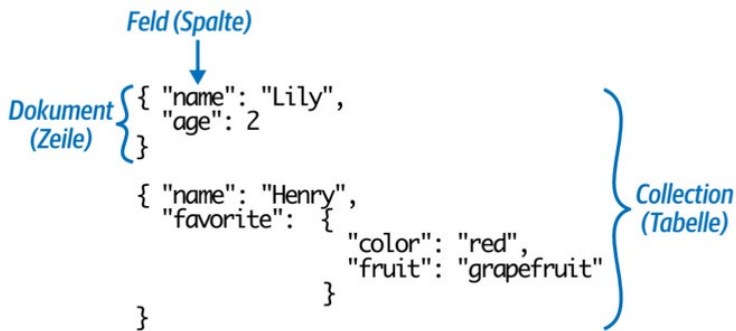


Abbildung 54 Datenstruktur (Zhao, 2022)

Eine Datenbank ist der Ort, indem Du Deine Dinge wie z. B. Bausteine oder Figuren aufbewahren kannst. Diese Datenbank oder “Sammlung” beinhaltet verschiedene Dinge wie Bausteine, Figuren usw., die wiederum über verschiedene Eigenschaften wie Farbe=Rot, Form=Würfel verfügen können. Was man in seiner Bausteindatenbank aufbewahrt und wie die Bausteine zueinander in Verbindung stehen, wird in Form des Entity-Relationship-Modells beschrieben.

Dinge (Entitäten)

Entitäten sind Dinge, über die wir Informationen sammeln möchten. In unserem Spiel sind dies die Bausteine oder auch Figuren. Jeder Baustein – zum Beispiel der grüne Stein mit der Form 2x3 gebogen oder die Figur Flamingo – stellen für sich eine Entität/Ding dar.

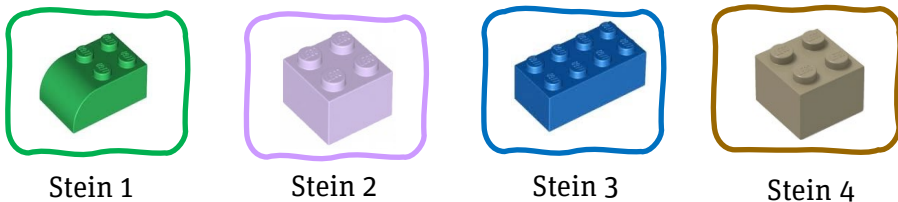


Abbildung 55: Bausteine in der Datenbank © Steininger

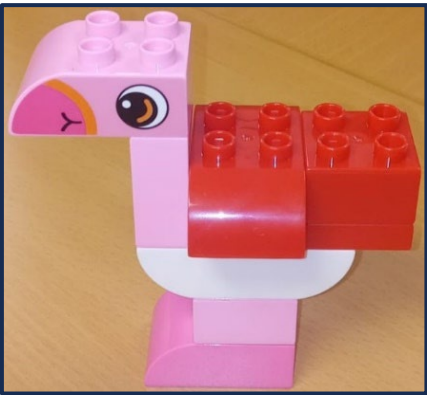


Abbildung 56 Flamingo aus Klemmbausteinen © Steininger

Eigenschaften

Jedes Ding kann über verschiedene Eigenschaften oder Merkmale verfügen. Zum Beispiel könnte unser Baustein die Eigenschaften Farbe und Form haben.



Baustein	
NUMMER	1
FARBE	rot
FORM	Stein 2x4 (Block)

Tabelle 5 Eintrag für einen Stein © Steininger

Verbindungen

Neben der Entität (Ding) also dem Baustein selbst spielt auch die Verbindung zu anderen Bausteinen oder Figuren eine große Rolle. Denn ein Baustein steht mit einer Figur in Verbindung oder anders ausgedrückt eine Figur wie z. B. der Flamingo benötigt einen oder viele Bausteine, um seine Gestalt anzunehmen.

Im Allgemeinen können zwischen drei wesentlichen Verbindungen unterschieden werden. Dies wird anhand des Spiels skizziert:

- **1:1 Verbindung**

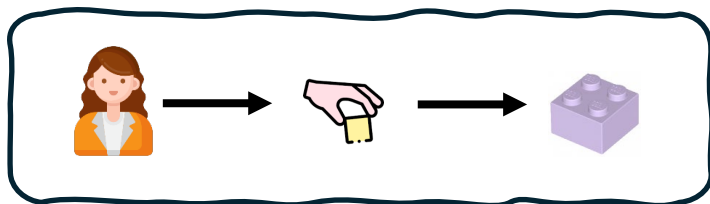


Abbildung 57 Beispiel 1:1 Verbindung © Steininger

Dies bedeutet beispielsweise, dass ein Schüler einen Baustein aus der Sammlung entnimmt oder daher eine Figur lediglich einen Baustein benötigen würde, um seine Gestalt anzunehmen.

- **1:n Verbindung (1: viele)**

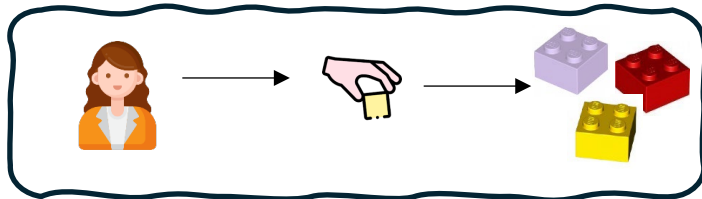


Abbildung 58 Beispiel 1:n Verbindung © Steininger

Dies trifft schon eher auf unser Spiel zu, denn das heißt, dass ein Schüler viele Bausteine aus der Bausteine-Datenbank entnimmt, um sein Muster auf der Karte zusammenzubauen. Das bedeutet daher auch, dass eine Figur viele Bausteine benötigt, um seine Gestalt anzunehmen.

- **m:n Verbindung (viele:viele)**

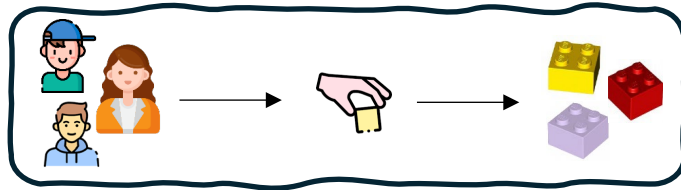


Abbildung 59 Beispiel m:n Verbindung

Auch das ist eine Verbindung, die in unserem Spiel zu-
trifft, denn viele Schüler aus der Klasse müssen viele
Bausteine aus der Datenbank entnehmen, um ihre Figu-
ren zusammenbauen zu können (=da viele Figuren auch
viele Bausteine benötigen, um ihre Gestalten anzuneh-
men).

Datenmodellierung, ER-Modell und Relationen

Die Datenmodellierung ist ein wesentlicher Faktor bei der Entwicklung eines Informatiksystems. Es muss ein Datenbank-Schema gefunden werden, sodass alle erforderlichen Daten im Datenbank-Schema abgespei-
chert werden können, die Konsistenz der Daten sichergestellt wird sowie schließlich effizient auf die Daten zugegriffen werden kann (vgl. Studer 2019).

Zur Beschreibung der Daten umfasst das ER-Modells folgende Elemente
– Entitäten bzw. Entitätstypen, Beziehungen und Attribute. Entitätstypen
stellen Aspekte der realen Welt auf einem abstrakten Niveau dar. Bei-
spielsweise könnte ein konkreter Entitätstyp ein bestimmter Kunde oder

auch ein bestimmter Artikel sein. Beziehungstypen spezifizieren den Zusammenhang zwischen den Entitäten. Das bedeutet zum Beispiel bei einem Onlineshop, dass Kund:innen auch verschiedene Artikel bestellen können. Attribute beschreiben Entitäten oder Beziehungen genauer. Beispielsweise können Kunden mitunter über die Attribute „Name“, „Anschrift“, „Kundenklasse“, etc. – verfügen und dadurch näher beschrieben werden vgl. (Gadatsch, 2017).

Relationen und ihre Tabellen lassen sich nicht isoliert betrachten. Zu Illustration lässt sich als Beispiel eine Bestellung anführen, die diese mit ihren Bestellpositionen und den Daten des Rechnungsempfängers in einer Beziehung stehen. Somit werden für eine Bestellung mindestens drei Relationen benötigt: eine für die Bestellung, eine für die Bestellpositionen und eine weitere für die Rechnungsempfänger vgl. (Ernst, et al., 2016).

Abhängig von den Kardinalitäten können die Relationen wie folgt unterschieden werden:

- | | |
|------------------|--|
| 1 zu 1 Beziehung | Zu einem Tupel gehört genau ein anderes Tupel. Beispiel: Person \leftrightarrow Kopf.
Oder zu einem Tupel gehört ein oder kein anderes Tupel (Konditional).
Beispiel: Person \leftrightarrow Reisepass |
| 1 zu N Beziehung | Zu einem Tupel gehört mindestens ein (≥ 1) anderes Tupel. Beispiel: Mitarbeiter \rightarrow Adresse. Oder zu einem Tupel gehören beliebig viele (≥ 0) andere Tupel. Beispiel: Ehepaar \rightarrow Kinder |
| M zu N Beziehung | Zu einem Tupel gehören beliebig viele (≥ 0) andere Tupel umgekehrt gilt dasselbe. Beispiel: Auto — Fahrer |

Abbildung 60: (Ernst, et al., 2016)

Zusätzliche Informationen erhalten Sie unter den Links:

- [https://de.wikipedia.org/wiki/Relation_\(Datenbank\)](https://de.wikipedia.org/wiki/Relation_(Datenbank))
- https://en.wikipedia.org/wiki/Entity-relationship_model

Nach (Stickel, 2013) soll nachfolgendes ER-Diagramm beispielhaft den Ausschnitt aus dem Schema eines Betriebs darstellen, wie es auch in der Praxis anzutreffen ist.

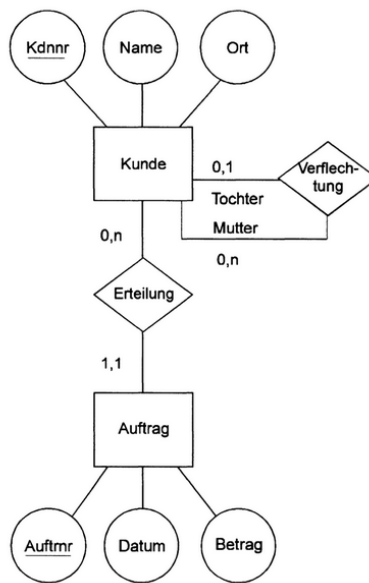


Abbildung 61 (Stickel, 2013)

Die Objekte (Entitäten) werden im Diagramm als Kästchen, die verschiedenen Beziehungsarten als Rauten und die Attribute als Ovale dargestellt. Dies bedeutet, dass Sie sämtliche Aspekte und Eigenschaften angeben, die Sie in der Datenbank speichern möchten. Ein unterstrichenes Attribut wie „Auftrnr“ repräsentiert einen Schlüssel wie den Primärschlüssel, der im nächsten Abschnitt genauer erläutert wird. Die Einträge wie 1,1 stehen für die jeweilige Kardinalität der Beziehungsart, die Sie bereits kennen gelernt haben. Beispielsweise würde die Kardinalität 0,n bei der Entität Kunde und der Beziehung Erteilung in diesem Fall bedeuten, dass jeder Kunde mindestens null oder mehrere Aufträge erteilt hat (vgl. ebd.).

Basisspiel (Primarstufe)

Eine Baustein-Datenbank ist jener Ort, wo man Bausteine aufbewahrt:



Abbildung 62: Bausteine – Pixabay.com

Wähle die Karte zu einer Figur und setze sie aus den Bausteinen der Datenbank zusammen:

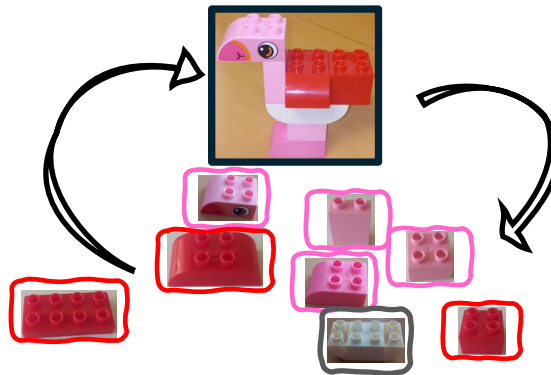


Abbildung 63 Figur und Bausteine © Steininger


Aufgaben

- Nimm einen Baustein aus der Bausteine-Sammlung (Datenbank).
- Beschreibe diesen Baustein.
- Zieh eine Karte.
- Baue das Muster auf der Karte.

Erweiterung 1 (Sekundarstufe 1)

Die Datenbank wird nun mit Tabellen erweitert:


Tabelle 6 Steine in der Block-Datenbank © Steininger

SteinId	Farbe	Form	Anzahl
	1 rot	Stein 2x2 (dünn)	2
	2 dunkelgrün	Stein 2x2 (dünn)	1
	3 hellblau	Stein 2x2 (dünn)	3
	4 hellgrün	Stein 2x2 (dünn)	10
	5 blau	Stein 2x2 (dünn)	4
	6 weiß	Stein 2x2 (dünn)	3
	7 gelb	Stein 2x2 (dünn)	2
	8 orange	Stein 2x2 (dünn)	2
	9 rot	Stein 2x2 (Würfel)	2
	10 dunkelblau	Stein 2x2 (Würfel)	2

Eine Tabelle (Tabelle 6) enthält die Bausteine mit einer Nummer (Identifikation) und Eigenschaften Farbe und Form. Die Anzahl gibt an, wie oft der Stein in der Datenbank vorkommt:

Die zweite Tabelle (Tabelle 7) beinhaltet die Figuren und die Anzahl der Bausteine, welche für deren Bau notwendig sind.


Tabelle 7 Figurentabelle mit Steinanzahl © Steininger

FigurId	Name	Anzahl Steine
	1 Flamingo	8
	2 Wal	10
	3 Elefant	14
	4 Eichhörnchen	9
	5 Schildkröte	8
	6 Krokodil	11
	7 Ente	8

Die dritte Tabelle beschreibt, welche Figur welche Steine enthält. Diese Tabelle verbindet die Steine mit den Figuren. Eine eindeutige Identifizierung einer Zeile kann man nur mit zwei Schlüsseln erreichen:

- Die **Id einer Figur** gepaart mit
- Der **Id eines Steins**

Tabelle 8 Steine zu Figur Verbindungstabelle © Steininger

FigurId	Name	SteinId	Anzahl pro Stein
	1 Flamingo	32	1
	1 Flamingo	9	1
	1 Flamingo	44	1
	1 Flamingo	17	1
	1 Flamingo	40	1
	1 Flamingo	25	1
	1 Flamingo	36	1

1	Flamingo	43	1
2	Wal	37	1
2	Wal	3	3
2	Wal	19	2
2	Wal	37	1
2	Wal	42	2
2	Wal	26	1

Datentypen

Die Daten in einer Datenbank können unterschiedliche Werte beinhalten. Beispielsweise kann die Baustein-Id eine Zahl oder der Name eine Zeichenkette sein. Dementsprechend sind in den Tabellen neben den Eigenschaften einer Entität (z. B. Baustein) auch der jeweilige Datentyp festzulegen. In der Baustein-Datenbank werden vor allem die folgenden Datentypen verwendet:

Tabelle 9: Datentypen¹

Datentyp	Beschreibung
INT	Steht für eine positive oder negative ganze Zahl
VARCHAR	Bedeutet ein Text variabler Länge
CHAR	Bedeutet eine bestimmte Textlänge
Weitere:	
FLOAT, DOUBLE	Wird für Gleitkommazahlen verwendet
DATE	Wird für Datumsangabe genutzt

Die „Id“ als „Primärschlüssel“

Wie ihr sehen könnt, enthält die Bausteine-Tabelle viele Zeilen. Es ist jedoch wichtig, die Datensätze (Zeilen) in einer Tabelle eindeutig zu

¹ Siehe <https://www.peterkropff.de> und <https://www.lucidchart.com>

kennzeichnen, um die Bausteine einfacher den Figuren zuzuordnen bzw. besser damit arbeiten zu können.



Abbildung 64: Primärschlüssel-eindeutige Identifizierung – Pixabay.com

Der Primärschlüssel der Bausteine-Tabelle bezieht sich auf die Eigenschaft Baustein-Id (Identifikationsnummer). Dieser dient also dazu, alle Bausteine vollständig voneinander zu unterscheiden. Zum Beispiel ist die Baustein-Id der Primärschlüssel für jeden Baustein in der Bausteine-Tabelle und die Figur-Id der Primärschlüssel der Figuren-Tabelle:

Tabelle 10 Steine mit Daten @ Steininger



SteinId 	Farbe	Form	Anzahl
1	rot	Stein 2x2 (dünn)	2
2	dunkelgrün	Stein 2x2 (dünn)	1
3	hellblau	Stein 2x2 (dünn)	3

Tabelle 11 Figuren mit Daten

Figur-Id 	Name	Anzahl Bausteine
1	Flamingo	8
2	Wal	10
3	Elefant	14

Mit diesen Schlüsseln können die Bausteine und die Figuren in der Datenbank eindeutig identifiziert bzw. erkannt werden.

Die Festlegung des Primär- und Fremdschlüssels sind die wichtigsten Integritätsbestimmungen in SQL vgl. (IBM, 2021):

- Der Primärschlüssel untersagt die Eingabe in einer oder mehreren Spalten innerhalb einer Tabelle doppelte Werte. Die Definition von Primärschlüssel gewährleistet, dass eine Spalte oder eine Kombination aus mehreren Spalten einen eindeutigen Wert für einen Datensatz beinhalten.
- Der Fremdschlüssel ist eine logische Regel über Werte in einer oder mehreren Spalten in einer Tabelle. Ein Fremdschlüssel ist ein Attribut, das in einer anderen Tabelle der Primärschlüssel ist wie z. B. im nachfolgenden Beispiel die AbteilungsNr. in der Tabelle Mitarbeiter. Hierdurch werden Verbindungen/Verknüpfungen zwischen Tabellen gewährleistet.

Nachfolgendes praxisnahes Beispiel zeigt Ihnen Tabellen die Werte enthalten, um Mitarbeiter und Abteilungen in einer Datenbank zu speichern:

PersonalNr.	Name	Gehalt	AbteilungsNr.
101	Meissner	75 000	1
102	Lehman	98 000	1
103	Kunze	75 000	2

AbteilungsNr.	Name	Standort
1	Einkauf	München
2	DV-Org.	Augsburg
3	Produktion	Straubing
4	Entwicklung	Augsburg
5	Verwaltung	München

Verknüpfung der Tabelle mit Mitarbeiter mit Abteilungen

PersonalNr.	Name	Gehalt	AbteilungsNr.	AName	Standort
101	Meissner	75 000	1	Einkauf	München
102	Lehman	98 000	1	Einkauf	München
103	Kunze	75 000	2	DV-Org.	Augsburg

Abbildung 65: In Anlehnung an (Ernst, et al., 2016)

Bei diesem Beispiel erfolgt eine Verknüpfung mit dem Attribut AbteilungsNr., welches in beiden Tabellen als Attribut vorhanden ist. In der Tabelle Abteilungen ist dies der Primärschlüssel und in der Tabelle Mitarbeiter der Fremdschlüssel. Dies führt dazu, dass die beiden Tabellen verknüpft werden können. Wie dies genau in der Erstellung der Tabelle deklariert wird, wird in Abschnitt 3 dargestellt.

Aufgaben

- Ziehe eine Karte.
- Trage in deiner Tabelle den Primärschlüssel und die Anzahl der jeweiligen Bausteine ein, die du für dein Muster benötigst.
- Baue deine Figur.

Erweiterung 2 (Sekundarstufe 2)

Um mit der Baustein-Datenbank kommunizieren zu können, müsst ihr eine spezielle Abfragesprache „Structured Query Language (SQL)“ anwenden.

Eine SQL-Abfrage ist eine Anweisung, um Daten aus der Baustein-Datenbank abzurufen oder zu bearbeiten. Mit diesen Abfragen kannst du beispielsweise auf die Tabelle Bausteine oder Figuren zugreifen, Bausteine filtern oder auch sortieren.

Nachfolgende Tabelle gibt dir einen Überblick über die wichtigsten SQL-Abfragen:

Tabelle 12: Klassische Sprachelemente in SQL¹

Abfrage	Beschreibung
SELECT (AUSWÄHLEN)	Mit SELECT kannst du die Daten aus einer oder mehreren Tabellen auswählen.
INSERT (EINFÜGEN)	Mit INSERT kann du neue Daten in eine Tabelle einfügen.
UPDATE (ÄNDERN)	Mit UPDATE kann du vorhandene Datensätze in einer Tabelle aktualisieren.
DELETE (LÖSCHEN)	Mit DELETE kann du Datensätze aus einer Tabelle löschen.
CREATE (ERZEUGEN)	Mit CREATE kannst du eine neue Tabelle oder Datenbank erstellen.
WHERE (FILTERUNG)	Mit WHERE kannst du die Datensätze auf bestimmte Bedingungen filtern.
JOIN (VERKNÜPFUNG)	Mit JOIN können Daten aus zwei oder mehreren Tabellen basierend auf einer gemeinsamen Spalte kombiniert werden.

In diesem Spiel benötigst du insbesondere die SQL-Abfrage „SELECT“, da du die richtigen Bausteine für dein Muster auswählen musst.

¹ Siehe <https://datascientest.com>

Die Abfrage sieht folgendermaßen aus:

SELECT (AUSWAHL)

Spaltenname

FROM (VON) TABELLE Tabellename;

Wenn du beispielsweise alle Bausteine aus der Bausteine-Tabelle auswählen möchtest, kannst du es so formulieren:

AUSWAHL * VON BAUSTEINEN;
(SELECT * FROM BAUSTEINE);

Das * bedeutet ALLE. Damit kannst du alle Bausteine aus dieser Tabelle auswählen. Du benötigst aber NICHT alle Bausteine, sondern nur bestimmte, um dein Muster zusammenzubauen. Dies bedeutet, dass du einen Baustein nach einer bestimmten Bedingung auswählen musst. Das kannst du mit einer WHERE-Bedingung durchführen.

Zum Beispiel benötigst du für den Flamingo den Baustein mit der Baustein-Id 1:

AUSWAHL Farbe, Form **VON** Bausteinen **WHERE (WO)** Stein-Id=1

Mit dieser Abfrage erhältst du nur den Stein mit der Id 1 und den Eigenschaften Farbe und Form.

Grundsätzlich lassen sich SQL-Anweisungen zwei wesentlichen Gruppen zuordnen – der DDL (Data Definition Language) und der DML (Data Manipulation Language). SQL umfasst Anweisungen, mit denen ein Datenbankschema wie etwa Tabellen etc. angelegt werden kann. Diese Anweisungen gehören zur DDL. Das Anlegen einer Tabelle wird beispielsweise mit CREATE TABLE durchgeführt, dies nachfolgend beispielhaft dargestellt wird (vgl. Ernst et al. 2016).

Dies bezieht sich wiederum auf die Tabelle Mitarbeiter des vorherigen Abschnitts:

```
CREATE TABLE Mitarbeiter (  
    PersonalNr.          int primary key,  
    Name                 varchar(255),  
    Gehalt               int,  
    AbteilungsNr.       int,  
    foreign key (AbteilungsNr.) references Abteilungen (AbteilungsNr)  
);
```

Zunächst muss die Tabelle mit Mitarbeiter benannt werden. Ebenso ist es erforderlich, zu jeder Spalte ein Name und ein Datentyp (siehe vorherigen Abschnitt) anzugeben. Eine Zeichenkette wird als varchar bezeichnet und darf in diesem Fall maximal 255 Zeichen lang sein. Darüber hinaus sind die Integritätsbedingungen (siehe vorherigen Abschnitt) zu definieren. Dazu gehören die PersonalNr. als Primärschlüssel und ferner die AbteilungsNr. als Fremdschlüssel vgl. (Ernst, et al., 2016).

Bei der Abfrage einer Datenbank werden Daten meist aus einen oder mehreren Tabellen abgerufen. Hierfür benötigen Sie die SELECT-Klausel, welche Spalten angibt, die Anweisungen für Sie wiedergeben (vgl. Zhao 2022). Beispielsweise:

```
SELECT PersonalNr., Name FROM Mitarbeiter;
```

Ergebnis der Abfrage:

PersonalNr.	Name
101	Meissner
102	Lehman
103	Kunze

Mit folgender Abfrage können Sie alle Spalten aus einer Tabelle auswählen:

```
SELECT * FROM Mitarbeiter;
```

Ergebnis der Abfrage:

PersonalNr.	Name	Gehalt	AbteilungsNr.
101	Meissner	75 000	1
102	Lehman	98 000	1
103	Kunze	75 000	2

Um Daten in eine Tabelle einzufügen, zu ändern oder zu löschen ermöglicht SQL die Ausführung von drei Anweisungen – INSERT, UPDATE und DELETE. Diese Anweisungen gehören zur DML innerhalb von SQL vgl. (Ernst, et al., 2016). Nennen wir das Beispiel von der Tabelle Mitarbeiter her und versuchen einen neuen Mitarbeiter einzufügen:

```
INSERT INTO Mitarbeiter VALUES (104, 'Meier', 45 000, 4);
```

Bereits vorhandene Datensätze können mit UPDATE verändert werden. Dabei ist die WHERE-Bedingung wesentlich, da sie die zu verändernden Zeilen identifiziert (vgl. ebd.).

```
UPDATE Mitarbeiter SET Gehalt=95 000 WHERE Name='Lehman';
```

Ebenso können Daten wieder gelöscht werden. Auch hier ist die WHERE-Bedingung erforderlich, um genau jene Daten zu löschen, auf diese die Bedingung zutrifft.

```
DELETE FROM Mitarbeiter WHERE Name='Meier';
```

Aufgaben

- Ziehe eine Karte.
- Trage in der Tabelle den Schlüssel, die Anzahl der Steine, sowie die Abfrage an die Datenbank ein.
- Baue deine Figur.

Apps & Services

Stefan Rappl, Harald Schwab , Stefanie Wieser

Web-APIs sind Anwendungsprogrammierschnittstelle über das World Wide Web für Anwendungen, um Daten zu empfangen, Datenbanken anzusprechen oder einfach nur um einfache Informationen wie eine Wetterprognose abzurufen. In diesem Workshop lernen Schüler:innen mit einer einfachen Fragen-Web-API die grundlegenden Funktionen des Webs zu verstehen und wie die Kommunikation mit einem Server funktioniert. Die Programmierung einer App und Verwendung der Fragen - API soll zeigen, wie Daten an den Server übermittelt werden und die Antwort vom Server an einen zurückgeschickt wird. Beim so genannten „Representational State Transfer“ (kurz REST) gibt es spezielle Operationen, um Anfragen durchzuführen, welche man im Zuge dieses Workshops kennen lernt. Der Workshop ist nur für die Sekundarstufe 2 vorgesehen und Vorwissen in Netzwerkkommunikation und dem Erstellen von Websites mit JavaScript Grundlagen ist Voraussetzung.

Dieser Workshop deckt diese Bereiche der Cloud Computing Lab Matrix ab:

Tabelle 13 Cola Matrix: Apps & Dienste © Knoll

Apps	Apps mit Cloud Anbindung (Firebase, Heroku, ...)		X	X
	Privatsphäre (z.B. Facebook als Cloud App)		X	X
Dienste	Dienste im Web (Cloud Ressourcen im Web)	X	X	X
	Web-API (Representational State Transfer)			X

Die Screenshots in diesem Kapitel stammen von der „Fragen-API“, welche unter der URL <https://cola.fh-joanneum.at/question-api/docs> aufgerufen werden kann!

Fragen API (Sekundarstufe 2)

Eine REST-API beruht auf genauen Designprinzipien und unterstützt webbasierte Apps in der Kommunikation miteinander. Grundsätzlich kann man eine REST-API zum Datenaustausch verwenden. Die API übermittelt die Anfragen an den Server und leitet die Antwort des Servers an dich weiter.¹

Auch die Fragen-App ist eine REST-API mit der man Anfragen an den Server schicken kann, um Daten zu erstellen, zu lesen, zu ändern oder zu löschen. Somit kann man einen ersten Einblick darüber bekommen, wie die Kommunikation zwischen dem Browser und der REST-API stattfindet:

CoLa2 FragenAPI

CoLa2 FragenAPI hilft dir mit einem einfachen Quiz die grundlegenden Funktionen des Web zu verstehen und wie die Kommunikation mit einem Server funktioniert. 🍷

Hauptseite

- Du kannst dir den **HTML** Code der Hauptseite anzeigen lassen.

Gehe zu

Fragen

- Du kannst dir alle **Fragen anzeigen** lassen, ...
 - oder du gibst im **zufall-Query** eine Zahl zwischen 1 und 100 ein, damit entsprechend viele Zufallsfragen abgerufen werden.
- Du kannst dir einzelne **Fragen** mit der **ID** anzeigen lassen.

Gehe zu

Gehe zu

Antwort

- Du kannst zu einer Frage eine **Antwort** zum Überprüfen an den Server senden.

Gehe zu

🏠 Zurück zur Cloud Computing Lab | Coding Lab Homepage [klick hier](#)

Weitere Infos und Kontakt

CoLa2 - Website
Send email to CoLa2
GNU AFFERO GENERAL PUBLIC LICENSE

Abbildung 66: QuizAPI unter <https://cola.fh-joanneum.at/question-api>

REST-APIs kommunizieren über HTTP (Hypertext Transfer Protocol). Mit der REST-API kannst du also Abfragen durchführen wie etwa Daten aus einer Datenbank lesen, Daten aktualisieren oder löschen. Das wird auch als CRUD-Operationen (Create, Read, Update, Delete) bezeichnet.

¹ Siehe <https://www.talend.com> und <https://www.ibm.com>.

Die Informationen können in jedem beliebigen Format geliefert werden wie z.B. HTML, Python oder der JavaScript Object Notation (JSON), welche man in dieser Übung kennen lernen wird. ¹

CRUD-Operationen

CRUD steht für Erstellen (Create), Lesen (Read), Aktualisieren (Update) und Löschen (Delete). Dies sind die grundlegenden Operationen, die man mit der REST-API durchführen kann. Im Rahmen der REST-API entsprechen sie den HTTP-Methoden POST, GET, PUT und DELETE². Im Rahmen des Workshops lernst man die Operationen POST und GET genauer kennen.

LESEN (HTTP GET)

Um Daten von der API abzurufen, kann man HTTP GET verwenden. Mit GET / ruft man die Hauptseite der Fragen-API ab.

Hauptseite Mit dieser Operation bekommst du die Startseite des Kurses als **HTML** zurück.

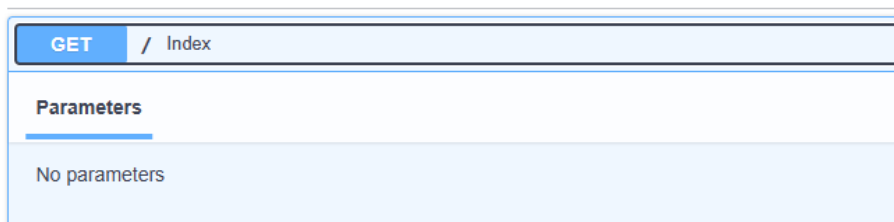


Abbildung 67: Lesender Aufruf der Hauptseite mit Hilfen

¹ Siehe <https://www.ibm.com>.

² Siehe <https://de.wikipedia.org/wiki/CRUD>

Mit GET /Fragen kann man sich alle oder eine zufällige Frage vom Server holen. Wenn man über das HTTP-Protokoll Daten (Ressourcen) vom Server anfordert, wird z.B. die Zahl 2 als GET-Parameter in der URL mitgeschickt. Dies sind Name-Wert-Paare. GET-Parameter beginnen immer mit einem Fragezeichen „?“. Darauf folgt der Name der Variable mit dem Wert, getrennt durch ein „=“.¹

Mit GET /Fragen/{id} kann man eine Frage mit einer speziellen Fragen-ID abrufen:

The screenshot shows a REST client interface for a GET request to the endpoint `/Fragen/{ID}` with the description "Quizfunktion". Under the "Parameters" tab, there is a table with two columns: "Name" and "Description". The table contains one entry: "ID" with a red asterisk and the text "required" next to it. Below the name, it specifies "integer" and "(path)". To the right of this information is a text input field containing the value "ID".

Name	Description
ID * required integer (path)	ID

Abbildung 68: Fragen mit einer bestimmten Id(entifikation) abrufen

The screenshot shows a REST client interface for a GET request to the endpoint `/Fragen` with the description "Operationen mit Fragen. Zum anzeigen von allen oder einzelnen Fragen. Quizfunktion". Under the "Parameters" tab, there is a table with two columns: "Name" and "Description". The table contains one entry: "zufall" with the text "(query)" below it. To the right of this information is a text input field containing the value "zufall".

Name	Description
zufall (query)	zufall

Abbildung 69: Zufällige Fragen holen

¹ Siehe <https://www.seobility.net/de/wiki/GET-Parameter>

ERSTELLEN (mit HTTP POST)

Um neue Daten in der API zu erstellen (wie in diesem Fall die Antwort hinzuzufügen), verwendet man die Methode HTTP POST. Zum Beispiel erstellt man die Antwort der Quizfrage mit POST und schickt diese an den Server.:

```
{
  ID: "1",
  Antwort: "Blauwahl"
}
```

Das Format, das in diesem Workshop verwendet wird, um die Antwort zu senden ist die JavaScript Object Notation (JSON). Dies ist für Menschen leicht zu lesen und zu schreiben und für Maschinen(=Computerprogramme) leicht zu analysieren und zu generieren¹. JSON wird verwendet, um Daten zwischen Computern auszutauschen. Die Syntax ist von der JavaScript-Objektnotation abgeleitet, jedoch ist das JSON-Format nur Text².

Weitere Informationen über JSON findest du unter:

- <https://www.json.org/json-en.html>
- https://www.w3schools.com/js/js_json_intro.asp.

¹ Siehe <https://www.json.org>

² Vergleiche mit <https://www.w3schools.com>

Antwort Um eine Frage zu beantworten, wird eine **Antwort** an den Server gesendet und überprüft.

POST /Antwort Quiz Answer

Parameters

No parameters

Request body required

Example Value | Schema

```
{
  "ID": "1",
  "Antwort": "Blaual"
}
```

Abbildung 70: Antwort mittels HTTP POST

Statusmeldungen und -codes

HTTP Status Codes geben dir Auskunft über das Ereignis einer Anfrage. Zu den gängigen Status-Codes gehören u.a.:

Tabelle 14: Statuscodes bei der Quiz API

Code	Beschreibung
200 (OK)	Erfolgreiche GET- oder PUT-Anfrage.
201 (CREATED)	Erfolgreiche POST-Anfrage.

204 (No Content)

Erfolgreiche DELETE-Anfrage.

400 (Bad Request)

Die Anfrage war ungültig oder konnte nicht zugestellt werden.

404 (Not Found)

Die angeforderte Ressource konnte nicht gefunden werden.

405 (Method not Allowed)

Die verwendete HTTP-Methode wird für diese Ressource nicht unterstützt.

Aufgaben

- Melde Dich bei der REST-API mit Namen ein.
- Ruf alle Fragen oder nur eine zufällige anhand der GET-Anfrage ab.
- Schicke Die Antwort via POST-Anfrage an den Server!
- Probiere ein paar Fragen aus!

Quiz App (Sekundarstufe 2)

Mit dem CoLa Editor beziehungsweise dem „Virtual Programming Lab“ (VCL) wurde in CoLa eine Webanwendung geschaffen, die es Schüler:innen erlaubt Apps zu schreiben. Im Rahmen von CoLa 2.0, dem „Cloud Computing Lab“ wurde der Editor nun erweitert, um die „Fragen-API“ verwenden zu können. Der CoLa-Editor/ das VCL ermöglicht nun eine vereinfachte Kommunikation mit der Web API, ähnlich wie die bereits die im Editor verwendete Bibliothek p5.js.

Funktionen

Die Bibliothek bietet vier Hauptfunktionen, um Daten von einem Webserver abzurufen. Alle Funktionen geben einen so genannten „Promise“ zurück. Ein „Promise“ ist ein Platzhalter für ein Ergebnis, das erst später verfügbar ist, zum Beispiel eine Antwort von einer Webseite. Man kann sagen: „Ich verspreche, dass ich dir irgendwann das Ergebnis gebe“, und man kann festlegen, was passiert, wenn es klappt (.then) oder schiefgeht (.catch).

In diesem Zusammenhang sind „async“ und „await“ nur eine bequemere Art, mit Promises zu arbeiten. „async“ macht eine Funktion automatisch zu einer Funktion, die ein Promise zurückgibt. „await“ pausiert die Ausführung dieser Funktion, bis das Promise fertig ist, und liefert direkt das Ergebnis, als wäre es synchron – man muss also nicht .then() verwenden.

Kurz: async/await macht asynchrone Abfragen wie GET oder POST einfacher und lesbarer und sollten daher hier verwendet werden.

- Funktion „**GET**“
führt einen GET-Request an die angegebene URI aus und liefert die Antwort als Zeichenkette.
- Funktion „**GET Json**“
führt einen GET-Request aus und liefert die Antwort als JSON. Wenn die Webseite kein JSON bereitstellt, wird ein Objekt im Format { "data": "response text" } zurückgegeben. Bei Fehlern erhält man ein Objekt wie:

```
{
  "error": "... reason ...",
  "status": 400,
  "statusText": "Client Side Error"
}
```


- Funktion „**POST**“
führt einen POST-Request mit einem Payload ¹(String oder Objekt) aus und liefert die Antwort als Zeichenkette.
- Funktion „**POST_Json**“
Entspricht „GET_Json“, führt jedoch einen POST-Request mit dem Payload aus.

Der folgende Code (Abbildung 71) fügt einer Webseite zuerst eigenes CSS hinzu, damit Listen grün und quadratisch aussehen. Er erstellt einen Bereich (div), in dem Inhalte angezeigt werden. Danach holt der Code einen Programmier-Witz von einer Webseite und zeigt ihn in einem Absatz (p) an. Außerdem lädt er Fragen von einer API, zeigt jede Frage als Absatz an und darunter die möglichen Antworten als Liste. Falls bei einem der Abrufe ein Fehler passiert, wird dieser in der Konsole angezeigt.

Die Funktion `customCSS(css)` erstellt ein neues `<style>`-Element, also einen kleinen Block für CSS-Designregeln. Danach setzt sie dieses Element ganz an den Anfang des `<body>` der Webseite. So kann man mit JavaScript eigenes CSS einfügen. Mit dem Aufruf von `customCSS(...)` wird die Funktion benutzt, um CSS-Regeln einzufügen. Es wird festgelegt, dass Listen (ul) grüne, quadratische Punkte bekommen. Das bedeutet: Sobald später Listen auf der Seite entstehen, sehen sie automatisch so aus.

Mit `“const out = document.createElement('div')”` wird ein neues `<div>`-Element erzeugt – ein leerer Bereich auf der Webseite, wie eine kleine Box. Mit `document.body.appendChild(out)` wird diese Box sichtbar gemacht, weil sie an die Webseite angehängt wird. In diese Box werden später alle Ergebnisse eingefügt.

¹ Ein Payload ist die Datenmenge, die man bei einer Anfrage an einen Server schickt, z. B. bei einem POST-Request. Kurz gesagt: Es ist der „Inhalt“ oder die „Nachricht“, die der Server verarbeiten soll.

```

function customCSS(css) {
    const el = document.createElement('style');
    document.body.insertBefore(el, document.body.firstChild);
}
customCSS(`
    ul {
        list-style: square inside;
        color: green;
    }
`);

const out = document.createElement('div');
document.body.appendChild(out);

// GET-Request Beispiel
GET('https://v2.jokeapi.dev/joke/Programming?for-
mat=txt&type=single')
    .then((text) => {
        const p = document.createElement('p');
        out.appendChild(p);
        p.innerText = text;
    });

// GET_Json Beispiel
GET_Json('https://cola.fh-joanneum.at/question-api/Fragen')
    .then((fragen) => {
        for (const frage of fragen) {
            const q = document.createElement('p');
            const ul = document.createElement('ul');
            out.appendChild(q);
            out.appendChild(ul);
            q.innerText = frage.Frage;
            for (const antwort of frage['Antwortmöglich-
keiten']) {
                const li = document.createElement('li');
                ul.appendChild(li);
                li.innerText = antwort;
            }
        }
    })
    .catch(err => console.error(err));

```

Abbildung 71 Beispiel zur Einbindung der Fragen-API in VCL

Der erste Aufruf `GET('https://v2.jokeapi.dev/joke/Programming?...')` holt einen Programmier-Witz von einem Webserver. Wenn die Antwort angekommen ist (`.then((text) => { ... })`), wird ein `<p>`-Element erstellt. Dieses bekommt den Text des Witzes und wird in die out-Box eingefügt. So erscheint der Witz auf der Webseite.

Der Aufruf `GET_Json('https://cola.fh-joanneum.at/question-api/Fragen')` ruft eine Liste von Fragen als JSON ab. Wenn die Daten da sind, macht der Code Folgendes:

- Für jede Frage wird ein `<p>` erzeugt, das nur Fragetext enthält.
- Direkt darunter wird eine ``-Liste erzeugt.
- Für jede Antwortmöglichkeit wird ein ``-Element erzeugt und der Liste hinzugefügt.

So entstehen auf der Seite die Frage und darunter eine Liste aller Antworten.

Fehlerbehandlung mit `.catch(err => console.error(err))` bedeutet:

Wenn irgendwo im Ablauf ein Fehler passiert (z. B. Server nicht erreichbar), wird er in der Browser-Konsole ausgegeben. Dadurch weiß man, was schiefgelaufen ist.

Aufgaben

- Starte das Visual Coding Lab.
- Hole über die vorgestellten Methoden eine Zufallsfrage und zeige sie auf der Webseite.
- Programmiere eine Logi, die es ermöglicht eine Antwort zu wählen- und zeige dann einen Indikator für Richtig oder Falsch!

Probiere ein paar Fragen aus! Erweitere Dein Programm nach Belieben!

Block API (Sekundarstufe 2)

Als Ergänzung zum Bereich Datenbanken und als Verbindung zu Apps & Services dient die Block-API, welche es ermöglicht, die Baustein-Datenbank über eine definierte Web-Schnittstelle anzusprechen.

Die folgende Grafik zeigt den Aufbau der Datenbank. Im oberen gelben Bereich stehen die Informationen über die Spieler:innen – also wer spielt und welche Bausteine diese Person besitzt. Im unteren rosa Bereich liegen die festen Grunddaten: welche Figuren es gibt, welche Bausteine existieren und aus welchen Bausteinen die Figuren bestehen. Die Linien zwischen den Tabellen zeigen die Beziehungen, also wer welche Bausteine hat und welche Bausteine zu einer Figur gehören. So lässt sich jederzeit nachvollziehen, welche User welche Elemente besitzen und wie alles zusammenhängt:

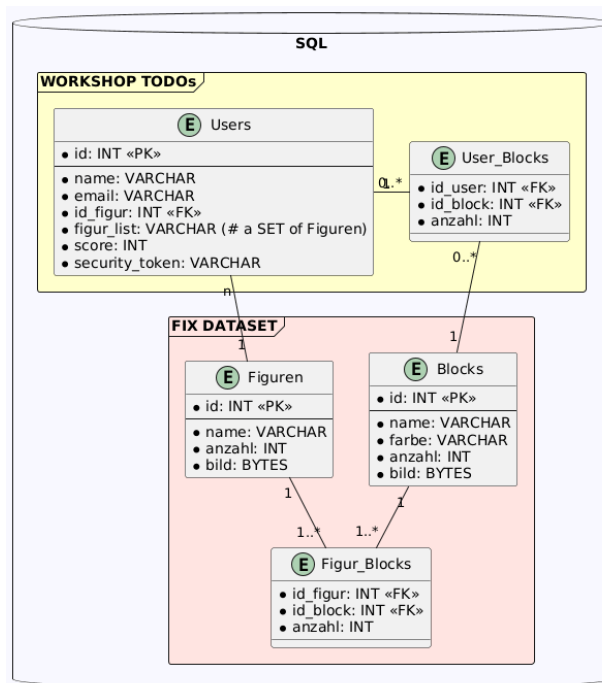


Abbildung 72 UML-Diagramm der Baustein-Datenbank © Rappi

Safer Internet & Cloud

Sicherheit im digitalen Raum

Sabine Proßnegg, Kaja Unger

Im Rahmen der rechtlichen Cola-Workshops wurde auf konkrete Fragen junger Nutzer:innen zum Thema: „Wie kann ich mich sicher im digitalen Raum bewegen“ eingegangen. Viele Schüler:innen machen sich Gedanken zu den Themen wie Privatsphäre, Sicherheit im Netz, etwas weniger oft um Urheberrechte oder Cybercrime. Gesetzestexte und juristische Fachliteratur sind mitunter keine leichte Lektüre, daher ist es wichtig, die für diese Zielgruppe bedeutsamen Aspekte so nahezubringen, dass Schüler:innen von den Vorträgen auch wirklich profitieren können. Zentral sind dabei unseres Erachtens Kompetenzen wie kritisches Denken, Selbstbewusstsein bei gleichzeitiger Offenheit für andere Ideen. Ganz wichtig ist das Wissen, wie und wo Unterstützung zu holen oder im Bedarfsfall auch Hilfe zu finden ist. Im Folgenden tauchen wir kurz in ein breites Spektrum an Themen ein und zeigen, wie diese Kompetenzen im Rahmen von Cola vermittelt wurden.

„Ge Cloud“

Im Workshop Ge Cloud, einem Wortspiel, in dem Medienkompetenz und Urheberrecht stecken, geht es in erster Linie um Konsumation. Hier braucht es kritisches Denken, wozu es auch gehört, sich zu überlegen, warum überhaupt etwas geschrieben oder gezeigt wird (oder eben nicht) und welche Absichten dahinterstehen. „Glaubenssätze“, die beinahe mantraartig - auch von gängigen Medien - verbreitet werden, sind kritisch zu hinterfragen. Gleichzeitig darf nicht alles, nur weil es im

Netz ist und mit „copy+paste“ leicht kopiert werden darf, für jeden Zweck verwendet werden.

Das Schüren von Ängsten und wie kritisches Denken helfen kann

Eines haben wir durch – damals noch Facebook, heute Meta – gelernt: je negativer die Stimmung des Publikums, umso länger wird konsumiert (Kramer, 2021). Es ist also das ureigenste Interesse der (sozialen) Medien, die bekanntlich nach Anzahl der verweilten Minuten, der Klicks etc. bezahlt werden, Angst zu verbreiten. Jede Minute länger auf diesen Medien bedeutet mehr Einnahmen für große Plattformen, ein Phänomen, das man als Plattformkapitalismus bezeichnet. Denn: es gibt nichts umsonst. Es ist heute nur schwer möglich, den ungeheuren Einfluss der weltweit agierenden Techgiganten zu unterschätzen, Meta ist hier nur davon (ORF, 2020).

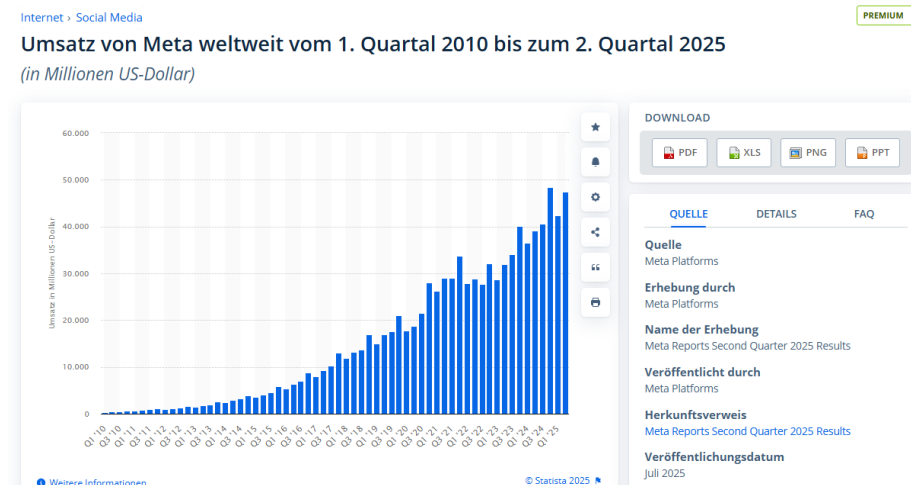


Abbildung 73 Statista 2025: Umsatz Meta

Gerade jungen Menschen wird permanent suggeriert, dass sie sich fürchten müssen: vor Krankheiten, vor Umweltkatastrophen, vor Kriminalität. Das ist doppelt problematisch, denn einerseits können junge

Menschen mit diesen Ängsten schwer umgehen¹ und andererseits können gerade sie nichts daran ändern, sind sie doch teilweise noch nicht einmal wahlberechtigt. In diesem Zusammenhang wird zwar viel über Fake News gesprochen und auf soziale Medien gezeigt, was zwar stimmt, aber zu kurz greift. Denn auch sogenannte Qualitätsmedien profitieren von diesen Ängsten und mischen immer öfter ihre Überzeugungen in vermeintlichen Qualitätsjournalismus. Zuletzt so geschehen bei der sehr renommierten BBC². Damit richten diese Qualitätsmedien aber doppelten Schaden an, denn wem soll man dann wirklich noch glauben?

Trotz des berühmten Ausspruchs, wonach man keiner Statistik glauben darf, die man nicht selbst gefälscht hat³, haben Statistiken doch einen großen Wert, wenn man sie sich zumindest selbst genauer ansieht und interpretiert. Um etwa das Mantra der steigenden Kriminalität anzusprechen, das derzeit bemüht wird, wohl auch um die Herabsetzung des Alters für Strafmündigkeit von derzeit 14 Jahren zu argumentieren, um nicht zu sagen: propagieren, zeigen wir gerne Statistiken von Statista Austria bzw. auch die Berichte des Bundeskriminalamtes und der Polizei. Demnach bleiben die Straftaten seit bereits fast 10 Jahren in etwa gleich. Sieht man sich weitere Zahlen an, erkennt man sogar, dass traditionelle Straftaten rückläufig sind. Was tatsächlich steigt, ist die Internetkriminalität (Cybercrime).

¹ So verwundert es nicht, dass die psychischen Probleme gerade bei jungen Menschen stark zunehmen, siehe auch die Statistik dazu unter Psychische Belastung bei jungen Menschen (Statista, 2024).

² Zum Beispiel der aktuelle Fall der BBC unter (Tagesschau, 2025).

³ Der Spruch wird meist Winston Churchill zugeschrieben.

Anzahl der angezeigten Straftaten in Österreich von 2014 bis 2024

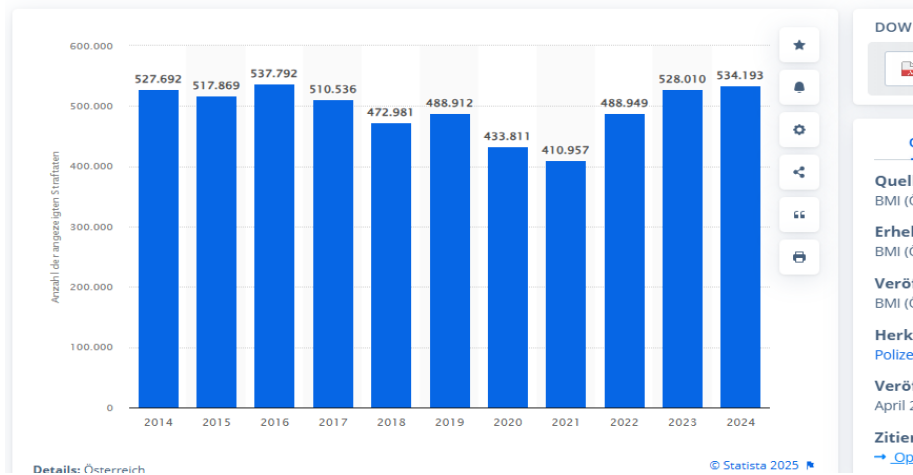


Abbildung 74 Statista 2025: Anzahl angezeigter Straftaten

Schutzrechte und was darf ich mir aus dem Netz nehmen

Gerade auf sozialen Medien ist es verlockend, Bilder, Videos und Musik hochzuladen. Das bringt uns zum Themenbereich der urheberrechtlichen Schutzrechte. Ein Foto, bei dem nichts dabeisteht, also keine Lizenzrechte angeführt sind, darf schlicht und einfach nicht für eine Präsentation, die eventuell auch noch auf der Webseite hochgeladen werden soll, verwendet werden. Das Urheberrecht schützt verschiedene Werke, etwa Sprachwerke (Werke der Literatur) wie Gedichte, Erzählungen, wissenschaftliche Artikel und wissenschaftliche Werke oder aber auch Computerprogramme, die unter diese Rubrik fallen. Das Urheberrecht schützt weiters Musik (Werke der Tonkunst), Werke der bildenden Künste wie Skulpturen, Fotografien, Grafiken, sowie Filmkunst und audiovisuelle Werke, womit Kinofilme, Dokumentationen und Fernsehserien gemeint sind.

Damit Schöpfer:innen der urheberrechtliche Schutz zukommt, ist es erforderlich, dass eine eigentümliche geistige Schöpfung einer natürlichen

Person vorliegt. Deshalb ist ein wichtiges Takeaway der Workshops: Respekt vor den Schöpfungen anderer Menschen zu haben und sich um entsprechende Genehmigungen, oder eben Lizenzen bzw. Nutzungsrecht, wie es juristisch heißt, zu kümmern. Plattformen wie pexels.com oder pixabay.com bieten genau das an: bei jedem Bild steht dabei, was damit gemacht werden darf und was nicht. Natürlich gibt es Ausnahmen, wie die Verwendung für rein private Zwecke, eine Parodie und ähnliches, aber jedenfalls gilt es, umsichtig zu sein.

Was passiert eigentlich mit meinen Personendaten

Das Thema Datenschutz ist inzwischen fest in unseren Gedanken verankert. Seit der Einführung der DSGVO im Mai 2018 sind sowohl der Datenschutz als auch die Privatsphäre feste Bestandteile der Überlegungen. Gerade junge Menschen sind hier wachsam in Bezug auf das „WAS“, also welche Daten sie von sich preisgeben und wie sie sich gegenüber anderen präsentieren. Umso bedauerlicher, eigentlich sogar völlig unverständlich, dass gerade diese europäische Stärke nun wieder rückgängig gemacht werden soll. Datenschutzaktivisten wie Max Schrems haben jahrelang für mehr Privatsphäre gegenüber den großen internationalen Plattformen gekämpft und damit vom EuGH auch Recht bekommen. Das sind Errungenschaften, die wir nicht leichtfertig aufgeben dürfen.

Persönliche Daten wie Wohnadresse, Telefonnummer oder Passwörter sollten immer geschützt werden. Die Nutzung von Nicknamen und die regelmäßige Kontrolle der Privatsphäre-Einstellungen helfen, die eigene Sicherheit zu erhöhen. Zum Thema Datenschutz gibt es auch viel Information und Hilfe im Netz, wie etwa die Plattform saferinternet.at.

Fehlendes Bewusstsein für Strafrecht

Während das Bewusstsein für Datenschutz oft groß ist, ist es beim Thema Cybercrime wenig ausgeprägt. Wird darüber in Schulen und zuhause vielleicht nicht so gerne gesprochen? Bei Sexualstraftaten im Cyberraum fehlt bei Kindern und Jugendlichen oft das Bewusstsein. Trends wie „Sexting“ werden als lustig und unproblematisch empfunden, sind es aber nicht. Die oft schwerwiegenden Konsequenzen, die die zum Spaß gemachten und übermittelten Bilder in weiterer Folge mit sich bringen können, werden von jungen Menschen schlicht unterschätzt. Werden die anzüglichen Bilder in weiterer Folge, etwa nach der Trennung oder nach einem Streit, nicht gelöscht, an Freund:innen weitergeleitet oder gar verbreitet, sind sich die Beteiligten der Konsequenzen meist nicht bewusst. Auch Cybermobbing wird oft nicht ernst genommen, und hier fehlt auch den Erwachsenen (Lehrenden, Eltern, Ansprechpersonen) das Wissen. Die Abhängigkeit von digitalen Medien, der Druck, dort positiv gesehen zu werden einerseits und die permanente Belästigung rund um die Uhr andererseits ist für Nicht-Digital-Natives, wie auch den Autorinnen, schlicht nicht vorstellbar. Es gibt an Schulen Initiativen wie Mental Health Days oder auch Plattformen wie AktivPräventiv, die Hilfe bieten. Schließlich muss niemand alles allein schaffen, daher keine falsche Scheu davor, sich dort zu melden.

Selbstbewusstsein und Offenheit

Eine negative Grundstimmung hilft den jungen Menschen nicht. Ganz im Gegenteil. Wir sind alle keine Passagiere, die in einem Zug sitzen und nichts tun können. Mit jeder Handlung und mit jedem Klick entscheiden wir, was passiert und auch, was nicht. Jede Minute, die wir nicht vor elektronischen Geräten verbringen, bedeutet, dass große Konzerne gerade nichts verdienen. Also, raus an die frische Luft und echte Menschen

treffen. Jede Frage, die wir selbst lösen, anstatt die KI zu fragen, bedeutet, dass unser Gehirn trainiert und die Umwelt geschont wird.

Letztlich geht es um nicht mehr oder weniger als um die Frage: wie wollen wir in Zukunft leben? Für uns ist das vielleicht nur mehr ein mehr oder weniger großer Teil unserer Lebenszeit, aber für Kinder und Jugendliche ist das sehr bedeutsam.



Abbildung 75 © Proßnegg, Unger, Pixabay.com

Conclusio

Kinder und Jugendliche brauchen nicht mehr oder strengere Strafen, sondern deutlich mehr Unterstützung und Hilfe, und zwar in vielerlei Hinsicht. Sie benötigen mehr rechtliches und technisches Wissen zu den hier genannten Themen, und konkrete Informationen zu ihren Rechten, den Dos and Don'ts. Dabei ist es wirklich wichtig, gerade im digitalen Raum, Vorsicht walten zu lassen, auch wenn vermeintliche Trends etwas Anderes propagieren.

Kinder und Jugendlichen brauchen mehr Unterstützung, um unabhängiger vom digitalen Raum zu werden. Leider ist die Lobbygruppe von „Mutter Natur“ nicht so finanzstark wie Google, Meta & Co. Tatsächlich gibt es aber erstaunlich viel Unterstützung und Hilfe, aufgesplittet auf eine

beinahe unübersichtliche Landschaft an Projekten, Organisationen und Initiativen. Lehrende, Eltern und andere Bezugspersonen können hier eine Hilfe sein, wenn sie sich dieses Wissen zuerst selbst aneignen. Es braucht also noch viel mehr Schulung und darum leistet gerade das Projekt Cola einen sehr wertvollen Beitrag.

Weiterführende Informationen:

bundeskriminalamt.at

www.polizei.gv.at

www.dsb.gv.at

saferinternet.at

<https://epicenter.works/>

Computer Chaos Club

None of your business noyb.at

Onlinesicherheit.at

www.rataufdraht.at

www.jugendinfo.at

www.jugendportal.at

www.stopline.at

www.gemeinsam-gegen-gewalt.at

www.schulpsychologie.at

www.plattformgegengewalt.at

www.antworten.at

www.mona-net.at

www.mentalhealthdays.eu

www.aktivpraeventiv.at

Zusammenfassung

Mathias Knoll

Die Workshops im Rahmen des CoLa- und des CoLa-2.0-Projekts sind das Herzstück der Bildungsinitiative dieser Projektreihe. Hier wird digitale Technologie für Schüler:innen wirklich greifbar – altersgerecht, praktisch, ein bisschen spielerisch aufbereitet.

Im Coding Lab tauchen die Schüler:innen zum ersten Mal in den kompletten Softwareentwicklungsprozess ein. Sie starten mit ersten Ideen, planen, setzen eigene kleine digitale Produkte um und sehen deren praktische Anwendung- wenn zum Beispiel eine Drohne erfolgreich einen Parcours meistert. Schritt für Schritt lernen sie die Grundlagen: Sie bauen Prototypen, überlegen, was Nutzer:innen brauchen, entwickeln im Team App- oder Spielideen und programmieren sie dann direkt in einer textbasierten Umgebung. Mit p5.js und Raspberry Pi erleben sie, wie Software entsteht. Dabei merken sie schnell, dass Programmieren viel mehr ist als nur Blöcke zusammenzuschieben – es ist ein kreativer Prozess, bei dem Problemlösung, logisches Denken und Ausprobieren im Mittelpunkt stehen. Die moderne Lernumgebung im Lab macht das Ganze noch inspirierender – Schulen schätzen diese Workshops und die Nachfrage ist ungebremsst.

Im Cloud Computing Lab geht es dann mit zentralen Technologien der IT-Welt weiter, die unter dem Synonym „Cloud“ zusammengefasst wurden. Jeder Workshop lehnt sich an eine Cloud-Computing-Matrix an, die diese Technologien kategorisiert und genau vorgibt, welche Inhalte je nach Schulstufe passen – von der Primarstufe bis zu den zwei Sekundarstufen. Ein wichtiger, zentraler Punkt ist das Thema Netzwerke: In einem spielerischen Modell sehen die Kinder, wie Computer miteinander Nachrichten austauschen, wie Adressen funktionieren und wie Protokolle aufgebaut sind. Mit der Analogie zur Post werden Begriffe wie Adressierung und

Transportmechanismen in Computernetzwerken plötzlich ganz anschaulich. Auch Server werden verständlich erklärt, oft mit einem Vergleich zu einem Restaurant: Der Server nimmt Bestellungen entgegen, bearbeitet sie und liefert sie aus. Mit Hilfe eines Kartenspiels können die Schüler:innen die Kommunikation zwischen Client und Server selbst nachspielen. Damit wird das Thema Netzwerke erweitert und es werden zusätzliche Mechaniken in der Netzwerkkommunikation beleuchtet.

Es gibt auch Workshops rund um das Thema Hardware. Passend zum Thema „Server“ werden diese in Workshops auseinandergenommen, die Einzelteile untersucht und beschrieben und dann wieder zusammengebaut. In einfachen Beispielen und Spielen zu elektrotechnischen Bauteilen (Logikgatter) erfahren die Schüler:innen, was Bits sind, wie Transistoren aufgebaut sind und wie ein Flip-Flop Schaltkreis Informationen festhält. Die Spiele „Turing Tumble“ und „Spintronics“ spielen dabei eine große Rolle – mit Kugeln und Schaltern, Zahnrädern und Ketten wird komplexe Logik und Speichertechnik sichtbar.

Datenbanken sind ebenfalls im Fokus: Mit einem Baukastensystem lernen die Schüler:innen, was „Entitäten“, „Relationen“ und „Attribute“ sind und wie sie zusammenhängen. Figuren- und Bausteinsets dienen als greifbare Beispiele für Tabellen, Primärschlüssel und den Beziehungen der Daten zueinander – so wird klar, wie Daten strukturiert gespeichert und abgerufen werden.

Diese ganze Infrastruktur bildet die Basis für Cloud-Dienste und Programmier-Schnittstellen über das Internet (Web-APIs). Die Schüler:innen erfahren, wie Anwendungen über das Web miteinander sprechen, wie Daten über das Hypertext Transfer Protokoll laufen und wie sie selbst Infos per Web-API abrufen oder verschicken können. Die im Projekt entwickelte „Fragen-API“ ist dabei ein praktisches und leicht verständliches Beispiel. Auch die „Block-API“ ist so eine Schnittstelle, welche Zugang zur bereits erwähnten Baustein-Datenbank gewährt. Die Integration der neuen Workshops in die bestehenden Werkzeuge des „Coding Labs“

öffnet Schüler:innen ein noch breiteres Spektrum zu vielfältigen Technologien. Diese integrierte Form der Wissensvermittlung eröffnet ein breiteres Spektrum an Technologien und schafft ein ganzheitliches Gesamtbild: Schüler:innen erkennen, wie die Bausteine hinter modernen Webanwendungen zusammenwirken – vom ersten Code über Netzwerke und Server bis hin zu Datenbanken, Speichersystemen und APIs. Das Projekt wächst damit über seine ursprünglichen Grenzen hinaus und zeigt den Lernenden auf anschauliche Weise, wie vielseitig, vernetzt und kreativ die Welt der Informatik wirklich ist.

Durch die Zusammenführung der Workshops aus dem „Coding Lab“ und dem „Cloud Computing Lab“ ist ein umfassenderes und deutlich größeres Bildungskonzept entstanden, das digitale Technologien nicht mehr isoliert, sondern im Zusammenhang vermittelt. Die neuen Inhalte fügen sich nahtlos in die bestehenden Werkzeuge des Coding Labs ein und erweitern sie um zentrale Aspekte moderner IT-Infrastrukturen. So entsteht ein Lernraum, in dem Schüler:innen nicht nur programmieren, sondern zugleich verstehen, wie Anwendungen in der Realität funktionieren, wie Daten fließen, wie Server kommunizieren und wie die Cloud als Fundament digitaler Dienste arbeitet.

Alle Workshops verfolgen einen gemeinsamen Ansatz: Die Schüler:innen sollen nicht nur die Theorie kennenlernen, sondern Technik wirklich erleben, anwenden und mit spielerischen Analogien durchdringen. Das macht neugierig, fördert Kreativität und gibt ihnen Selbstvertrauen im Umgang mit digitaler Technologie. Die Workshops von CoLa und CoLa-2.0 und dem zukünftigen CoLa-3.0 zeigen eindrucksvoll, wie man selbst schwierige IT-Themen verständlich machen und gleichzeitig richtig Lust auf Informatik wecken kann.

Glossar

3D | Three Dimensional

Ein dreidimensionaler Raum – verwendet für Spiele, Modelle oder Drohnensteuerung.

AI | Artificial Intelligence

Der englische Begriff für Künstliche Intelligenz.

API | Application Programming Interface

Hilft Programmen, miteinander zu sprechen und Informationen auszutauschen.

CPU | Central Processing Unit

Der „Denker des Computers – führt Befehle aus und verarbeitet Daten.

CRUD | Create, Read, Update, Delete

Die vier Grundaktionen in einer Datenbank: erstellen, lesen, ändern, löschen.

CSS | Cascading Style Sheets

CSS ist eine Sprache, mit der man bestimmt, wie eine Webseite aussieht – also Farben, Schriftarten, Abstände und die ganze Gestaltung

DHCP | Dynamic Host Configuration Protocol

Ein Dienst, der Geräten im Netzwerk automatisch eine IP-Adresse gibt.

DNS | Domain Name System

Übersetzt Webseiten-Namen in Zahlenadressen, die Computer verstehen.

ER-Modell | Entity-Relationship-Modell

Ein Plan, der zeigt, wie verschiedene Daten miteinander verbunden sind.

FFG | Österreichische Forschungsförderungsgesellschaft

Sie unterstützt Projekte wie CoLa und CoLa 2.0, damit neue Ideen und Technologien entstehen können und Wissen vermittelt wird.

HTTP | Hypertext Transfer Protocol

Regelt, wie dein Browser mit einem Server spricht, um Webseiten anzuzeigen.

HTTP GET | Eine Get Anfrage

Der Standardaufruf, um Daten von einer Webseite abzurufen.

HTTP POST | Eine Post Anfrage

Eine Art Nachricht, mit der ein Browser Daten an einen Server schickt, z. B. bei einem Login.

IP | Internet Protocol

Gibt jedem Gerät im Internet eine eindeutige Adresse, damit Nachrichten ankommen.

JS | JavaScript

Eine Programmiersprache, die viele Webseiten interaktiv macht.

KI | Künstliche Intelligenz

Computerprogramme, die aus Beispielen lernen und selbst Entscheidungen treffen.

NIST | National Institute of Standards and Technology

Eine US-Behörde, die viele technische Standards – auch für Cloud Computing – definiert.

OSI-Modell | Open Systems Interconnection Model

Beschreibt in mehreren Schichten, wie Daten durch ein Netzwerk reisen.

p5.js | Processing for JavaScript

Eine Programmierbibliothek, mit der Schüler:innen Spiele und Animationen im Browser erstellen können.

RAM | Random Access Memory

Der Kurzzeitspeicher des Computers – speichert Dinge, die gerade gebraucht werden.

REST | Representational State Transfer

Eine Art, wie moderne Web-APIs aufgebaut sind, damit sie leicht nutzbar sind.

RPI | Raspberry Pi

Ein kleiner, günstiger Lerncomputer, der im Coding Lab verwendet wird.

SQL | Structured Query Language

Eine Sprache, um Datenbanken zu befragen und zu steuern.

TCP/IP | Transmission Control Protocol / Internet Protocol

Die beiden wichtigsten Regeln, wie Daten im Internet übertragen werden.

URL | Uniform Resource Locator

Die Internetadresse einer Webseite.

VR | Virtual Reality

Ein digitaler Raum, in den man mit einer Brille eintauchen kann.

Literaturverzeichnis

Anon., 2023. *What are CRUD operations in a REST API?*. [Online]
Available at: <https://www.basedash.com/blog/what-are-crud-operations-in-a-rest-api>

Bredenfeld, A. & Leimbach, T., 2010. *The Roberta Initiative*. s.l., s.n., pp. 558-567.

Ernst, H., Schmidt, J. & Beneken, G., 2016. *Grundkurs Informatik: Grundlagen und Konzepte für die erfolgreiche IT-Praxis*. s.l.:Springer Fachmedien Wiesbaden.

FFG, 2024. *Die Österreichische Forschungsförderungsgesellschaft FFG!*. [Online]
Available at: <https://www.ffg.at/>
[Zugriff am 17 01 2024].

Gadatsch, A., 2017. *Datenmodellierung für Einsteiger*. s.l.:Springer Fachmedien Wiesbaden.

Horváth, G. & Menyhárt, L., 2014. *Teaching introductory programming with JavaScript in higher education*. s.l., s.n., pp. 339-350.

IBM, 2021. *Typen von Integritätsbedingungen*. [Online]
Available at: <https://www.ibm.com/docs/de/db2/10.5?topic=constraints-types>
[Zugriff am 14 02 2025].

Kahn, K., 1996. ToonTalk™ — an animated programming environment for children. *Journal of Visual Languages & Computing*, Band 7, pp. 197-217.

Knoll, M., 2025. *Coding Lab / Cloud Computing Lab*. [Online]
Available at: <https://cola.fh-joanneum.at/workshops-cloud-computing-lab/>
[Zugriff am 01 11 2025].

Knoll, M. & Wieser, S., 2025. *Coding Lab / Cloud Computing Lab*.
[Online]

Available at: <https://cola.fh-joanneum.at/>
[Zugriff am 15.11.2025].

Krainz, E., Schwab, H., Niederl, F. & Kotal, K., 2022. *LEARNING-BY-CODING – CHALLENGES AND EXPERIENCES IN ESTABLISHING A CODING LAB FOR CHILDREN*. s.l., s.n., pp. 6010-6015.

Kramer, K., 2021. Warum schlechte Nachrichten anziehen. *Frankfurter Allgemeine Zeitung (FAZ)*, Konsum schlechter Nachrichten: Warum wir Doomscrolling mögen (FAZ 08.03.2021).

Lukas, G., 1972. *Uses of the LOGO programming language in undergraduate instruction*. s.l., s.n., pp. 1130-1136.

Maloney, J. et al., 2010. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, Band 10, pp. 1-15.

McCarthy, L., Reas, C. & Fry, B., 2015. *Getting started with P5.js: Making interactive graphics in JavaScript and Processing*. s.l.:Maker Media, Inc..

Mell, P. & Grance, T., 2011. *NIST Technical Series Publications*. [Online]
Available at:
<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>
[Zugriff am 3.11.2025].

ORF, 2020. *Ö1 auf orf.at*. [Online]
Available at: <https://oe1.orf.at/programm/20200612/601264/1-Wem-gehoren-die-fuenf-Tech-Giganten-2-Digitale-Lehren-der-Pandemie>
[Zugriff am 21.11.2025].

Schneider, H. et al., 2020. *IT-Qualifikationen für die österreichische Wirtschaft*, Wien: s.n.

Shylesh, S., 2017. *A Study of Software Development Life Cycle Process Models*, s.l.: s.n.

Slany, W., 2012. *Catroid: a mobile visual programming system for children*. s.l., s.n., pp. 300-303.

Statista, 2024. *Statista*. [Online]

Available at:

<https://de.statista.com/statistik/daten/studie/1469341/umfrage/haeufigkeit-von-psychischen-belastungen-unter-jungen-menschen/>
[Zugriff am 21.11.2025].

Stickel, E., 2013. *Konzeptuelle Datenmodellierung*. s.l.:Vieweg+Teubner Verlag.

Tabet, N., Gedawy, H., Alshikhabobakr, H. & Razak, S., 2016. *From Alice to Python: Introducing text-based programming in middle schools*. s.l., s.n., pp. 124-129.

Tagesschau, 2025. *Tagesschau*. [Online]

Available at: <https://www.tagesschau.de/ausland/europa/bbc-ruecktritte-100.html>
[Zugriff am 21.11.2025].

Tanenbaum, A., Feamster, N. & Wetherall, D., 2024. *Computernetzwerke*. 6., aktualisierte Auflage Hrsg. s.l.:Pearson Studium (Reihe: IT).

UpperStory, 2024. *Turing Tumble - Leitfaden für Pädagogen*. s.l.:UpperStory.

Zhao, A., 2022. *SQL – kurz & gut*. s.l.:dpunkt.verlag.

Abbildungsverzeichnis

Abbildung 1: Cloud Computing Lab Matrix, © Knoll.	9
Abbildung 2: Stufenweiser Workshop Aufbau, © Knoll.	11
Abbildung 3: Entwickelte Apps auf der Site, © Knoll, FlatIcon.com	12
Abbildung 4 Workshop Sujet © Knoll	14
Abbildung 5: Geräte, © Knoll.	16
Abbildung 6: Ausdrucke, Anleitungen & Verbrauchsmaterial, © Knoll.	16
Abbildung 7: Briefumschläge in drei Größen, © Knoll.	17
Abbildung 8 Symbol Netzwerkkarte - Flaticon.com	17
Abbildung 9: Symbole, eindeutige Markierungen von Netzwerkkarten © Knoll	18
Abbildung 10 Symbol Switch - Flaticon.com	18
Abbildung 11: Beispiel für die Kernfunktionalität eines Switchs @Knoll	19
Abbildung 12 Symbol Router - Flaticon.com	19
Abbildung 13 Bilder von PixaBay.com	21
Abbildung 14: Rollenzuteilung: Gerät zu Person © Knoll, PixaBay.com	22
Abbildung 15: Das Netzwerk der Gruppe "Grün" © Knoll, FlatIcon.com, PixaBay.com	23
Abbildung 16: Nachrichten zwischen physischen Komponenten werden über deren eindeutige Kennung adressiert, © Knoll	24
Abbildung 17 https://freesvg.org/caramel-candy	25
Abbildung 18: Susi und Peter kommunizieren © Knoll, FlatIcon.com, PixaBay.com	26
Abbildung 19: Nachrichten zwischen Personen mit logischer Adressierung über Namen und Gruppenfarbe © Knoll	27
Abbildung 20: Verschachtelung © Knoll	28

Abbildung 21:Karten von Peter und Susi, auf welchen ihre Informationen vermerkt sind. © Knoll.....	29
Abbildung 22 Gesamtnetzwerk © Knoll	30
Abbildung 23: Nachricht über Netzwerkgrenzen hinweg © Knoll.....	31
Abbildung 24: Zusatzkarten der Vermittler Holger und Liz © Knoll.....	32
Abbildung 25: Susi sendet eine Nachricht an Barbara © Knoll.....	33
Abbildung 26: Holger leitet an Liz, Vermittlerin von Gruppe Blau, weiter. © Knoll.....	33
Abbildung 27: Liz leitet die Nachricht in ihrem Netzwerk an Barbara weiter. © Knoll	34
Abbildung 28: Umschlag mit Adressierung an einen Server © Knoll.....	35
Abbildung 29: Links verschiedene „Zuckerl“-Server und rechts das entsprechende entwickelte Protokoll, © Knoll.....	36
Abbildung 30: Adressfelder aller Briefe, um eine Süßigkeit von einem "Zuckerlserver" zu beziehen. © Knoll.....	37
Abbildung 31: Das Netzwerkspiel im realen Kontext (Teil 1), © Knoll.....	39
Abbildung 32: Das Netzwerkspiel im realen Kontext (Teil 2), © Knoll.....	40
Abbildung 33: DHCP: Stühle und Spieler:innen © Schwab, FlatIcon.com,	46
Abbildung 34: DHCP: Stühle nummeriert, Spieler:innen wählen für sich eine Stuhlnummer © Schwab, FlatIcon.com	46
Abbildung 35: DHCP: Auswahl der Stühle ohne ein System © Schwab, FlatIcon.com	47
Abbildung 36: DHCP: Der Sitzplatz Manager, FlatIcon.com.....	47
Abbildung 37 (c) Schwab, FlatIcon.com	48

Abbildung 38 DHCP: Kein Chaos mit dem Sitzplatz-Manager © Schwab, FlatIcon.com	48
Abbildung 39 Beispielsrunde © Schwab, FlatIcon.com	53
Abbildung 40 Spielverlauf: Anfrage © Schwab, FlatIcon.com.....	54
Abbildung 41 Spielverlauf: 404 Not Found © Schwab, FlatIcon.com	55
Abbildung 42: Anfrage mit GET	56
Abbildung 43: Anfrage mit POST	57
Abbildung 44 Kompass , PixaBay.com.....	60
Abbildung 45 Analogie, wie Transistoren funktionieren	61
Abbildung 46 Logische Operationen: F(alse) und T(rue)	61
Abbildung 47 AND-Gatter: Nur beide Knöpfe ergeben Licht (Wahr) Bilder © Knoll	62
Abbildung 48 OR-Gatter: Jeder einzelne oder alle Knöpfe ergeben Licht (Wahr) Bilder © Knoll	62
Abbildung 49 Flip-Flop Schaltung © Knoll.....	63
Abbildung 50 Flip-Flop - 2 Diode einschalten © Knoll.....	63
Abbildung 51 Flip Flop: Diode ausschalten © Knoll.....	64
Abbildung 52 Flip Flop in Turing Tumble (UpperStory, 2024)	65
Abbildung 53 Tabelle (Zhao, 2022).....	67
Abbildung 54 Datenstruktur (Zhao, 2022).....	68
Abbildung 55: Bausteine in der Datenbank © Steininger.....	68
Abbildung 56 Flamingo aus Klemmbausteinen © Steininger	69
Abbildung 57 Beispiel 1:1 Verbindung © Steininger	70
Abbildung 58 Beispiel 1:n Verbindung © Steininger	70
Abbildung 59 Beispiel m:n Verbindung.....	71
Abbildung 60: (Ernst, et al., 2016)	72
Abbildung 61 (Stickel, 2013)	73
Abbildung 62: Bausteine – PixaBay.com	74
Abbildung 63 Figur und Bausteine © Steininger	74
Abbildung 64: Primärschlüssel-eindeutige Identifizierung – PixaBay.com	78
Abbildung 65: In Anlehnung an (Ernst, et al., 2016).....	79

Abbildung 66: QuizAPI unter https://cola.fh-joanneum.at/question-api	86
Abbildung 67: Lesender Aufruf der Hauptseite mit Hilfen.....	87
Abbildung 68: Fragen mit einer bestimmten Id(entifikation) abrufen.....	88
Abbildung 69: Zufällige Fragen holen	88
Abbildung 70: Antwort mittels HTTP POST.....	90
Abbildung 71 Beispiel zur Einbindung der Fragen-API in VCL.....	94
Abbildung 72 UML-Diagramm der Baustein-Datenbank © Rappl	96
Abbildung 73 Statista 2025:Umsatz Meta	98
Abbildung 74 Statista 2025: Anzahl angezeigter Straftaten	100
Abbildung 75 © Proßnegg, Unger, PixaBay.com.....	103

Tabellenverzeichnis

Tabelle 1 Cloud Matrix: Netzwerke, © Knoll.....	15
Tabelle 2 Cola Matrix: Server © Knoll	41
Tabelle 3 Cloud Matrix: Speicher © Knoll	59
Tabelle 4 Cloud Matrix: Datenbanken © Knoll.....	66
Tabelle 5 Eintrag für einen Stein © Steininger	69
Tabelle 6 Steine in der Block-Datenbank © Steininger	75
Tabelle 7 Figurentabelle mit Steinanzahl © Steininger.....	76
Tabelle 8 Steine zu Figur Verbindungstabelle © Steininger	76
Tabelle 9: Datentypen	77
Tabelle 10 Steine mit Daten © Steininger	78
Tabelle 11 Figuren mit Daten	78
Tabelle 12: Klassische Sprachelemente in SQL.....	81
Tabelle 13 Cola Matrix: Apps & Dienste © Knoll.....	85
Tabelle 14: Statuscodes bei der Quiz API	90

IT+

FH | JOANNEUM
University of Applied Sciences



IT+

FH JOANNEUM
University of Applied Sciences



IT+

FH JOANNEUM
INSTITUT Software Design und Security
Werk-VI-Straße 46
8605 Kapfenberg, AUSTRIA



[fh-joanneum/iit](https://fh-joanneum.at/iit)