2nd BMBF Big Data All Hands Meeting

2nd Smart Data Innovation Conference

11.10.2017

Technischer Datenschutz von personenbezogenen Daten

Matthias Gabel

Karlsruher Institut für Technologie (KIT)


SaaS
Office 365
Dropbox
IaaS / PaaS
amazon
web services
Microsoft®
SQL Azure™

Vorteile Cloud Computing

- Ressourceneffizienz
- Kosteneffizienz
- Skalierbarkeit
- Verfügbarkeit
- kein eigenes Rechenzentrum notwendig

- … was sind die Nachteile?

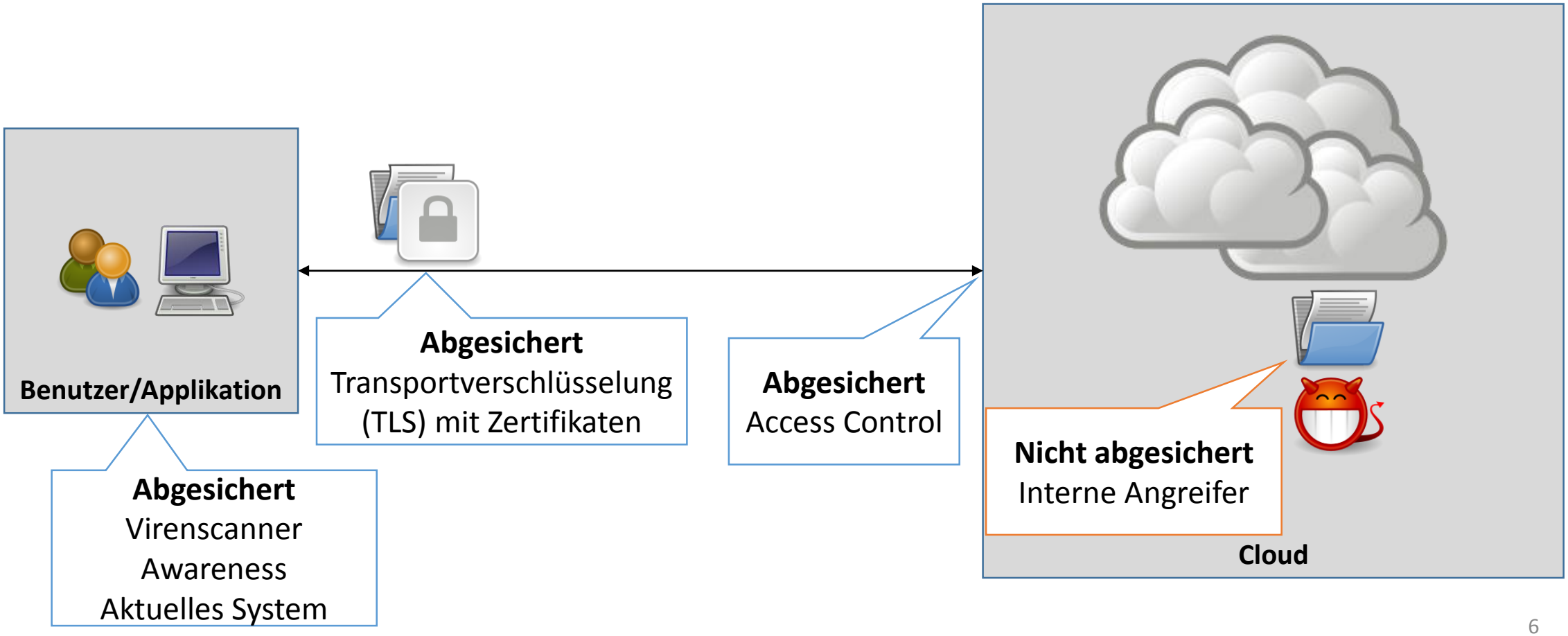
Angreifertypen

- extern
  - nicht authentifiziert
- Intern
  - hat berechtigten Zugang oder Zutritt (nach Anlage zu §9 BDSG)
    - Reguläre Nutzer
    - Admins
    - Reinigungskraft/Hausmeister
- Passiv
  - Daten lesen

- Aktiv
  - Daten schreiben

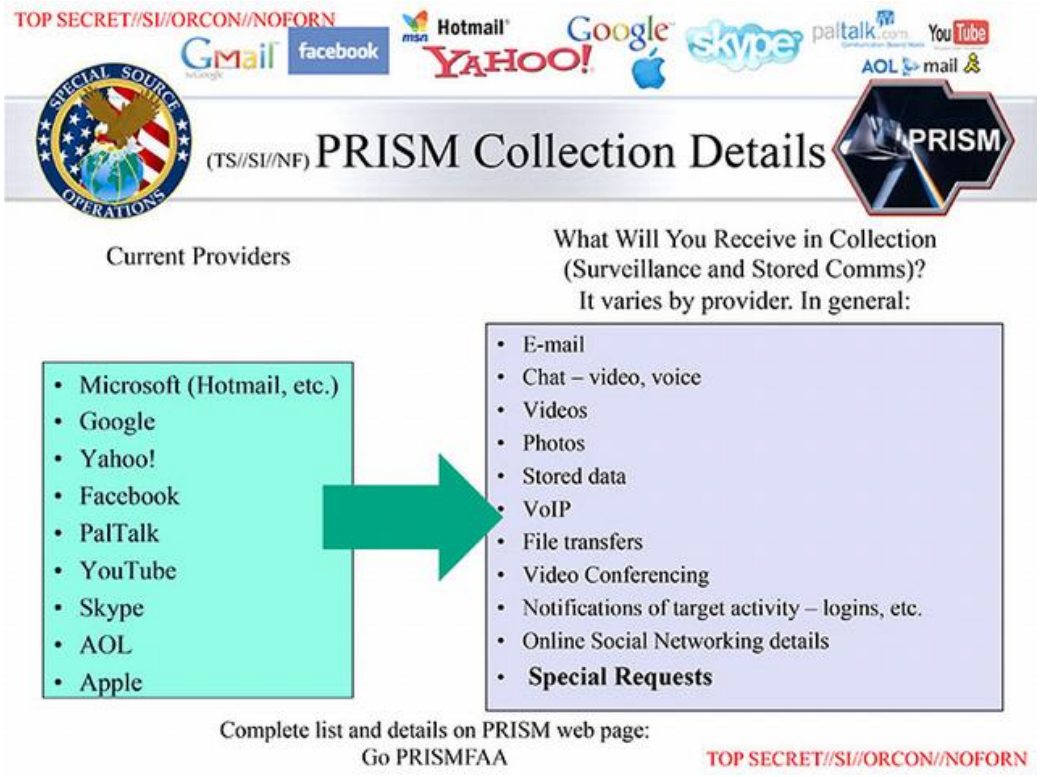
Interne Angreifer

- Heutiges Problem: Meist nur Schutz gegen externe Angreifer

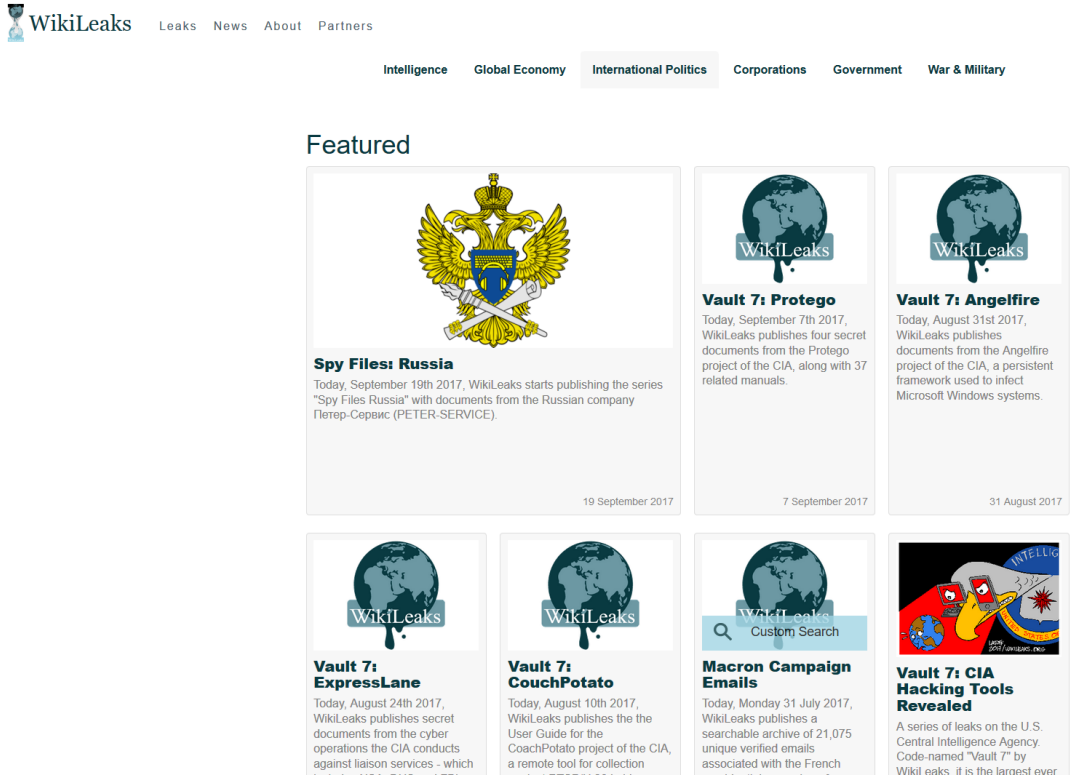


Interne Angreifer: Beispiele

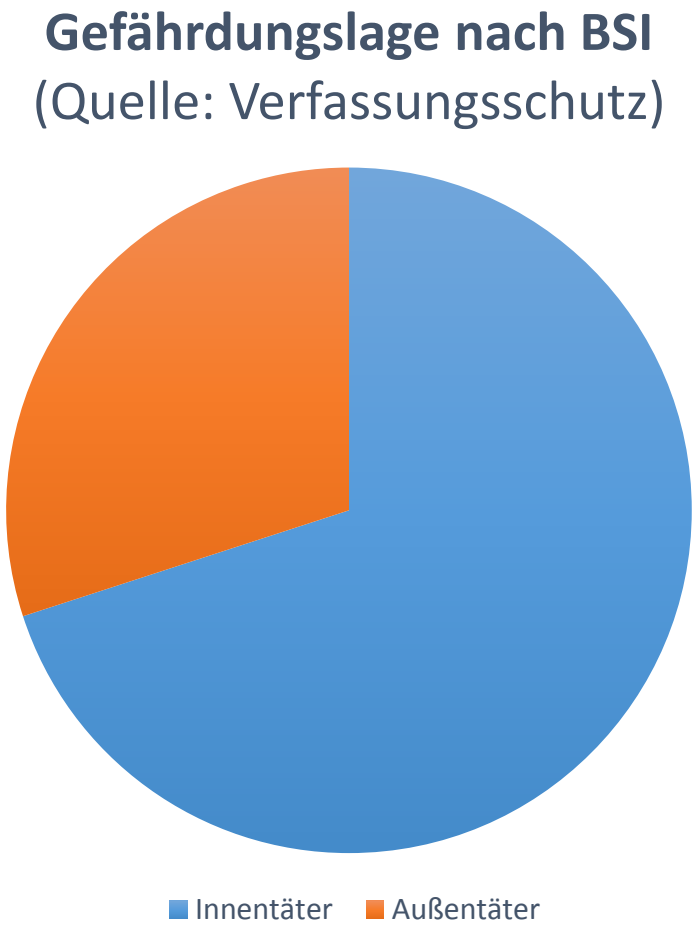
PRISM



WikiLeaks

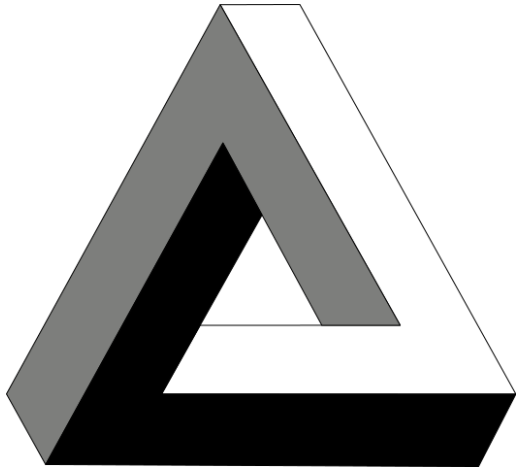


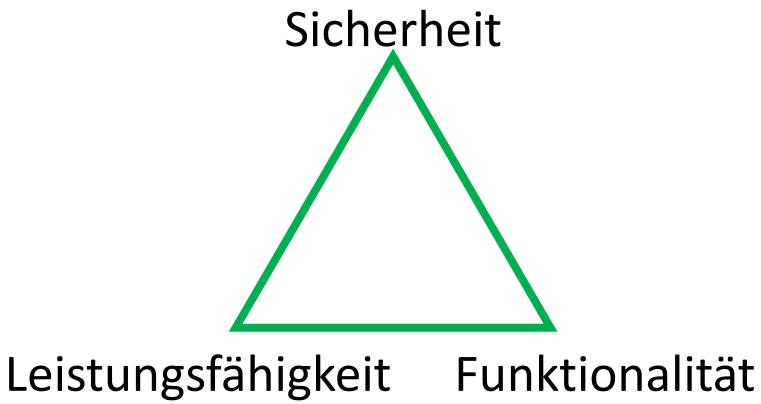
Interne Angreifer: Gefährdungslage



Cloud-Computing-Paradoxon

- Cloud-Anbieter benötigt lesbare (unverschlüsselte) Daten
- Cloud-Anbieter darf so wenig wir möglich aus Daten lernen





Klassische Lösungsansätze

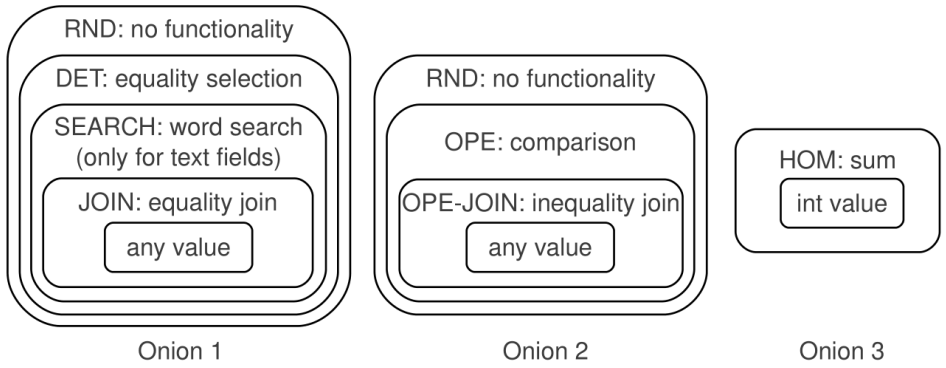
- FHE: vollhomomorphe Verschlüsselung [Gentry2009]
  - Sehr hohe Sicherheit (Semantic Security)
  - Zu langsam (Mehraufwand von 10^7 , d.h. 4 Monate statt 1 Sekunde [Gentry2015])

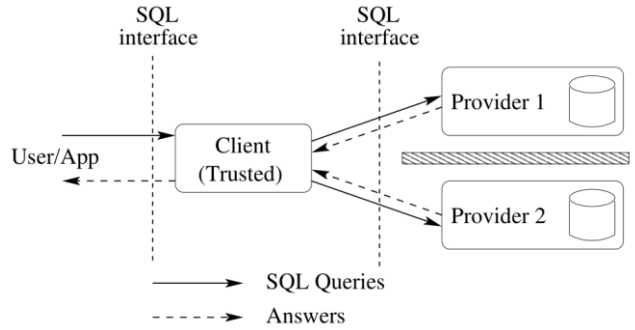
- MPC: Sichere Mehrparteienberechnung [Goldreich1987]
  - Nicht allgemein anwendbar
  - Zu langsam (Mehraufwand von 10^6)

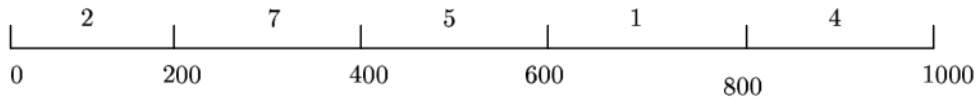
- Alles verschlüsseln
  - Nur für reine Datenablagen (Backups) geeignet

Spezielle Ansätze

- CryptDB (MIT CSAIL, 2011) [Popa2011]
  - Praxistauglich und performant
  - Sicherheit abhängig vom Anwendungsfall
  - Keine Sicherheitsgarantie
- Reine Datenverteilung [Aggarwal2005]
  - Keine Verschlüsselung
  - Keine Sicherheitsgarantie
- Grobe Indizes [Hacigümüs2002]
  - Ineffizient
  - Keine Sicherheitsgarantie

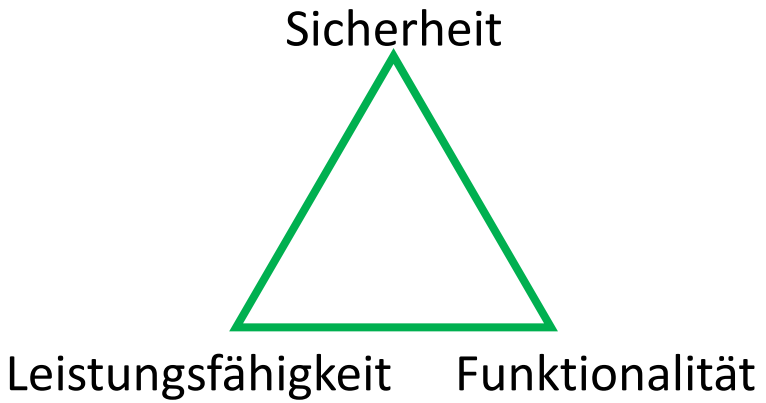






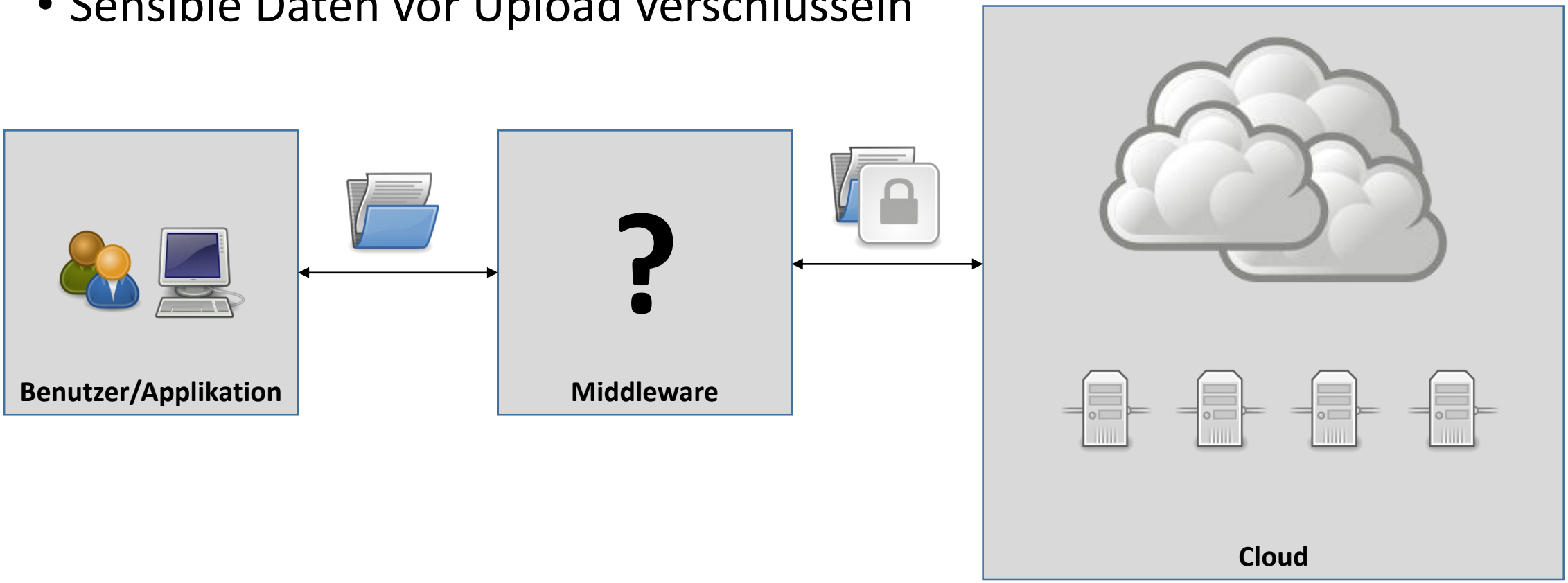
Anforderungen

- Sicherheit
  - Angemessener Schutz gegen passive Angreifer
- Leistungsfähigkeit
  - Sublineare Suche
  - Vertretbarer Aufwand
- Funktionalität
  - Unterstützung einer hinreichenden Menge von SQL-Queries
  - Transparenz gegenüber Benutzer/Applikation



Ansatz

- Einfügen einer Zwischenschicht
- Sensible Daten vor Upload verschlüsseln



Existierende Sicherheitsbegriffe

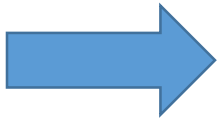
- k-Anonymity
- l-Diversity
- t-Closeness
- Differential Privacy

… sind alle nicht geeignet

Beispiel: k-Anonymität

ZIP Code	Gender	DOB	Condition
02139	m	1970/05/14	Heart Disease
02138	f	1978/10/11	Obesity
02139	f	1987/11/01	Flu
02139	f	1961/02/11	Diabetes
02142	f	1962/01/29	Cancer
02141	m	1964/08/05	Cancer
02142	m	1968/09/18	Cancer
...	...	...	...

Beispielhafte Patientendatenbank



Generalisierung

ZIP Code	Gender	DOB	Condition
021**	*	1970-1989	Heart Disease
021**	*	1970-1989	Obesity
021**	*	1970-1989	Flu
021**	*	1960-1969	Diabetes
021**	*	1960-1969	Cancer
021**	*	1960-1969	Cancer
021**	*	1960-1969	Cancer
...	...	...	...

3-anonyme Version der obigen Tabelle

Fragestellungen

- Cloud-Computing-Paradoxon
- Klassische Ansätze zu ineffizient

- => Gibt es schwächere Sicherheitsbegriffe, die verwendbar sind?
- => Gibt es dazu effiziente Verfahren?

Definition IND-ICP

- IND-ICP = Indistinguishability Under Independent Column Permutation
- Spielbasierter Sicherheitsbegriff
- Angreifer versucht zwei Varianten einer Tabelle nach der Transformation Enc(·) zu unterscheiden
- In jeder Spalte darf permutiert werden
- Beispiel:

d_0		
A	1	x
B	2	y
C	3	z

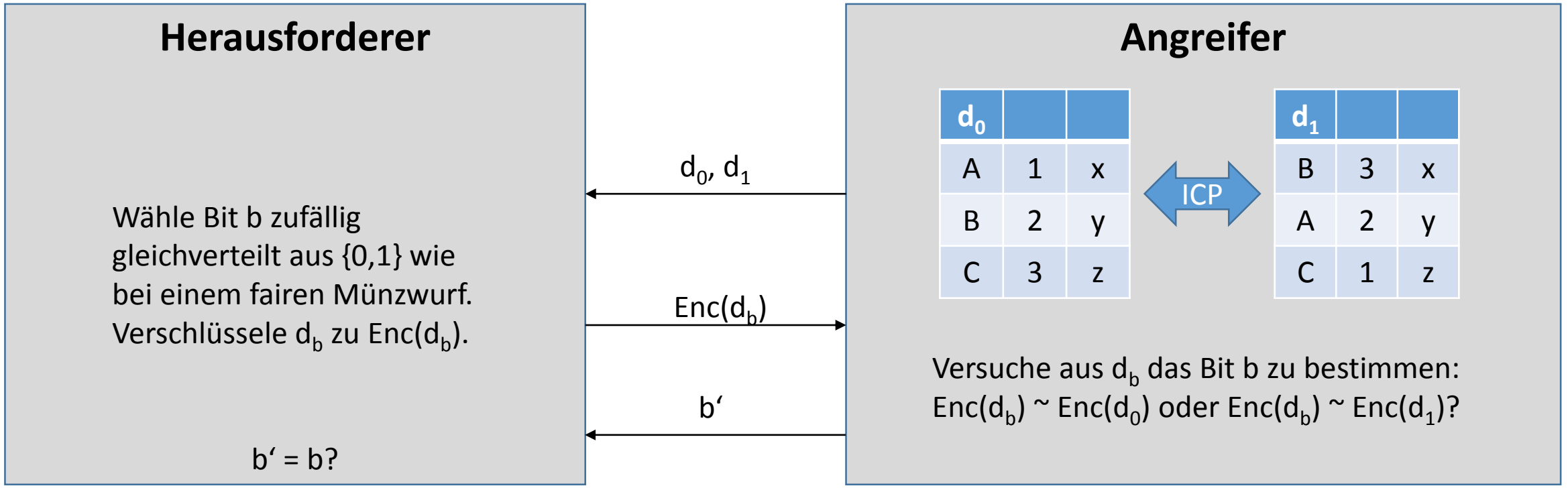
ICP

d_1		
B	3	x
A	2	y
C	1	z

~~ICP~~

d_2		
x	1	A
B	2	y
C	z	3

Sicherheitsbegriff IND-ICP



Intuition: Angreifer kann zwischen $Enc(d_0)$ und $Enc(d_1)$ nicht unterscheiden
=> Enc(·) versteckt Assoziation der Attribute

Transformation Enc(·)

ID	Vorname	Nachname	Geburtstag	Stadt
1	Frank	Maier	1.2.1980	Karlsruhe
2	Adam	Schulz	1.2.1990	Berlin
3	Hannes	Schneider	1.2.2000	Hamburg
4	Frank	Schmidt	1.2.2010	München

Original

ID	Encrypted Data
1	Enc(Frank, Maier, 1.2.1980, Karlsruhe)
2	Enc(Adam, Schulz, 1.2.1990, Berlin)
3	Enc(Hannes, Schneider, 1.2.2000, Hamburg)
4	Enc(Frank, Schmidt, 1.2.2010, München)

Daten

Keyword	IDs
Vorname:Frank	Enc(1, 4)
Vorname:Adam	Enc(2)
Vorname:Hannes	Enc(3)

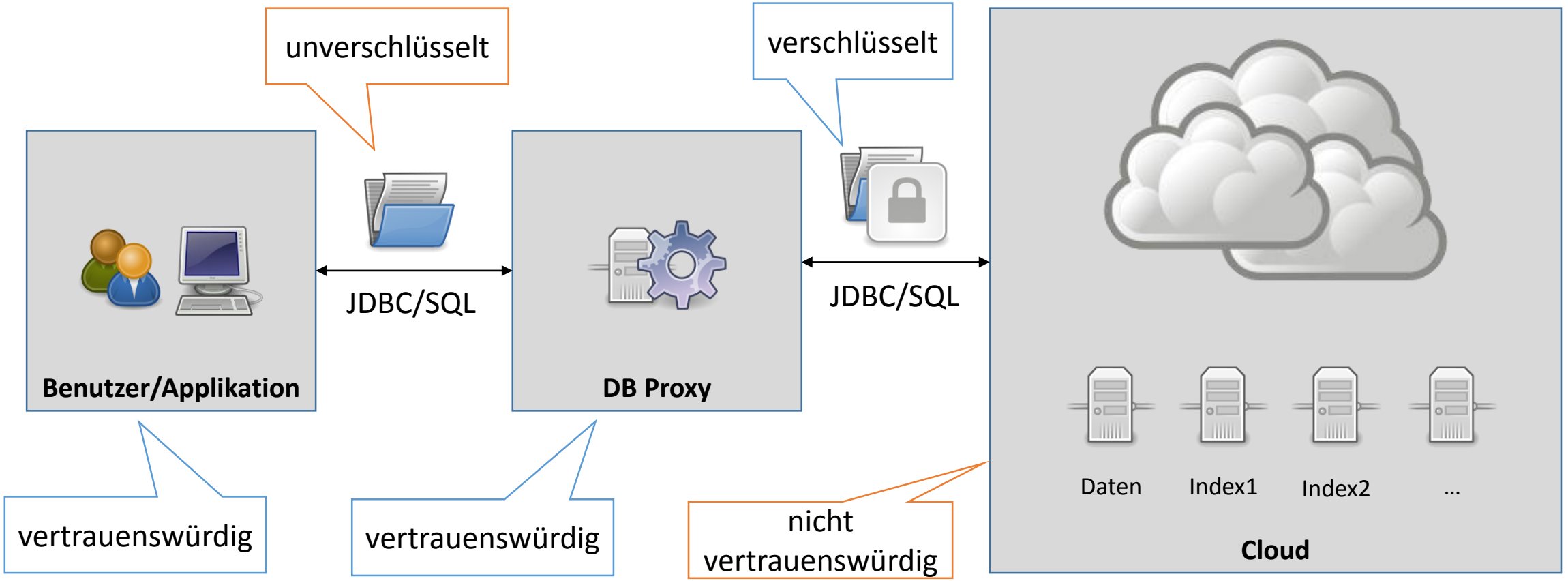
Index 1

Keyword	IDs
Nachname:Maier	Enc(1)
Nachname:Schulz	Enc(2)
Nachname:Schneider	Enc(3)
Nachname:Schmidt	Enc(4)

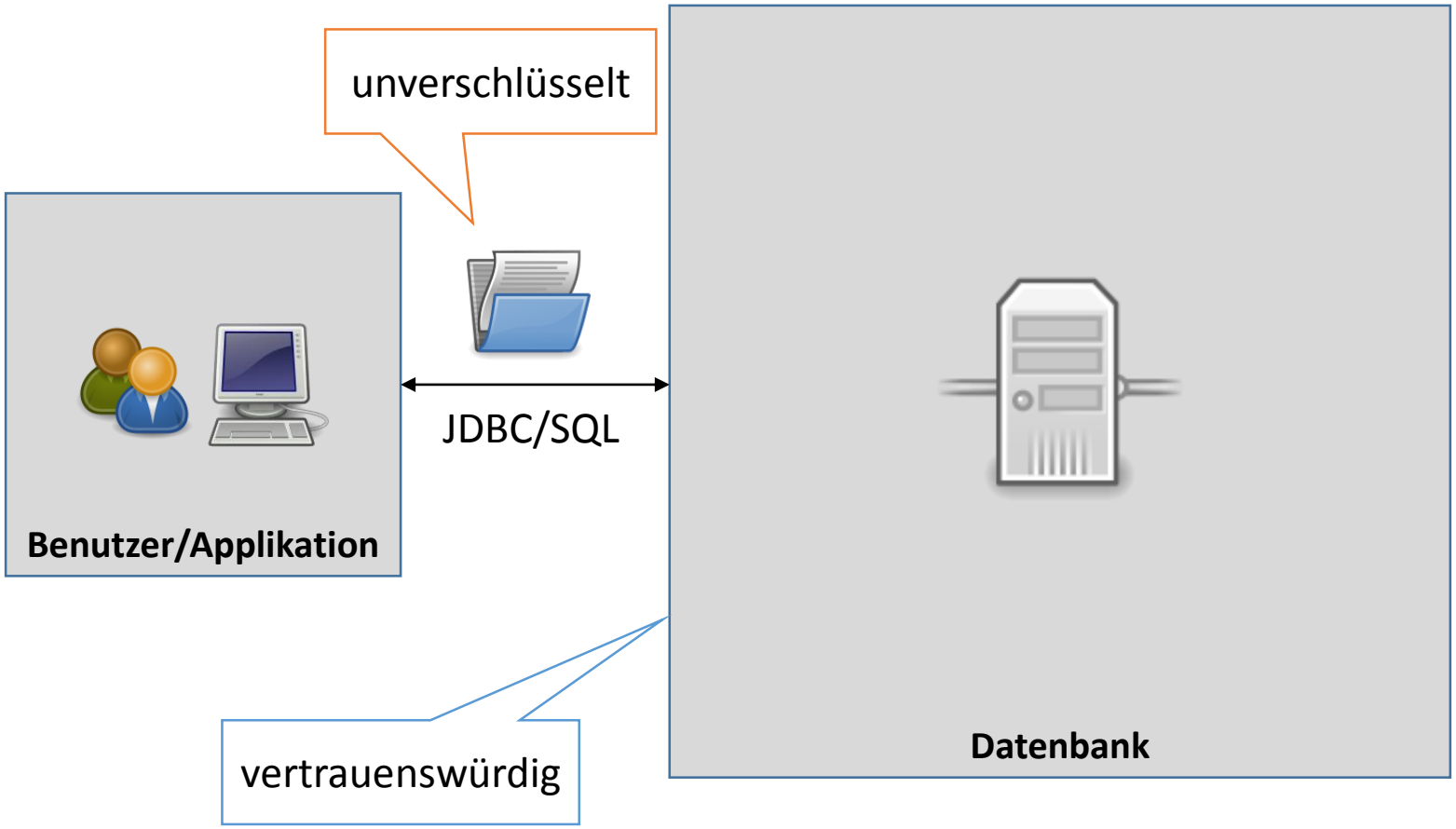
Index 2

Enc(Original)

Architektur



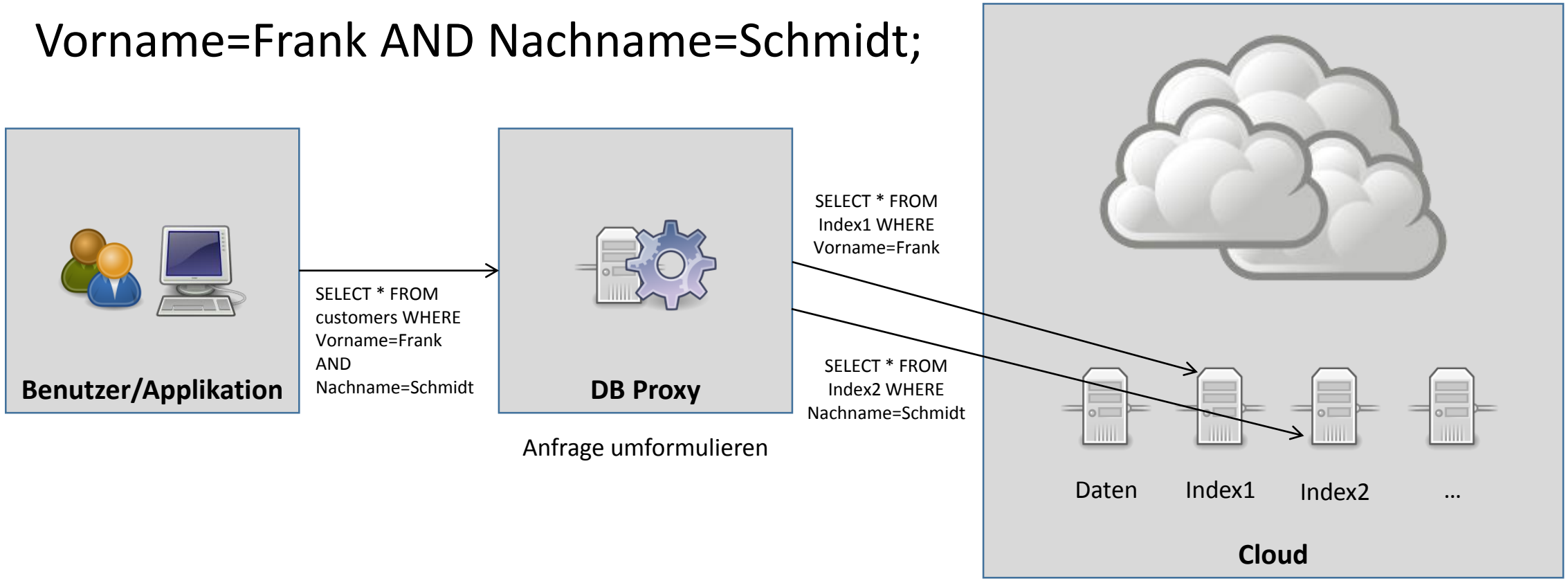
Architektur (Nutzersicht)



Beispiel 1/4

ID	Vorname	Nachname	Geburtstag	Stadt
1	Frank	Maier	1.2.1980	Karlsruhe
2	Adam	Schulz	1.2.1990	Berlin
3	Hannes	Schneider	1.2.2000	Hamburg
4	Frank	Schmidt	1.2.2010	München

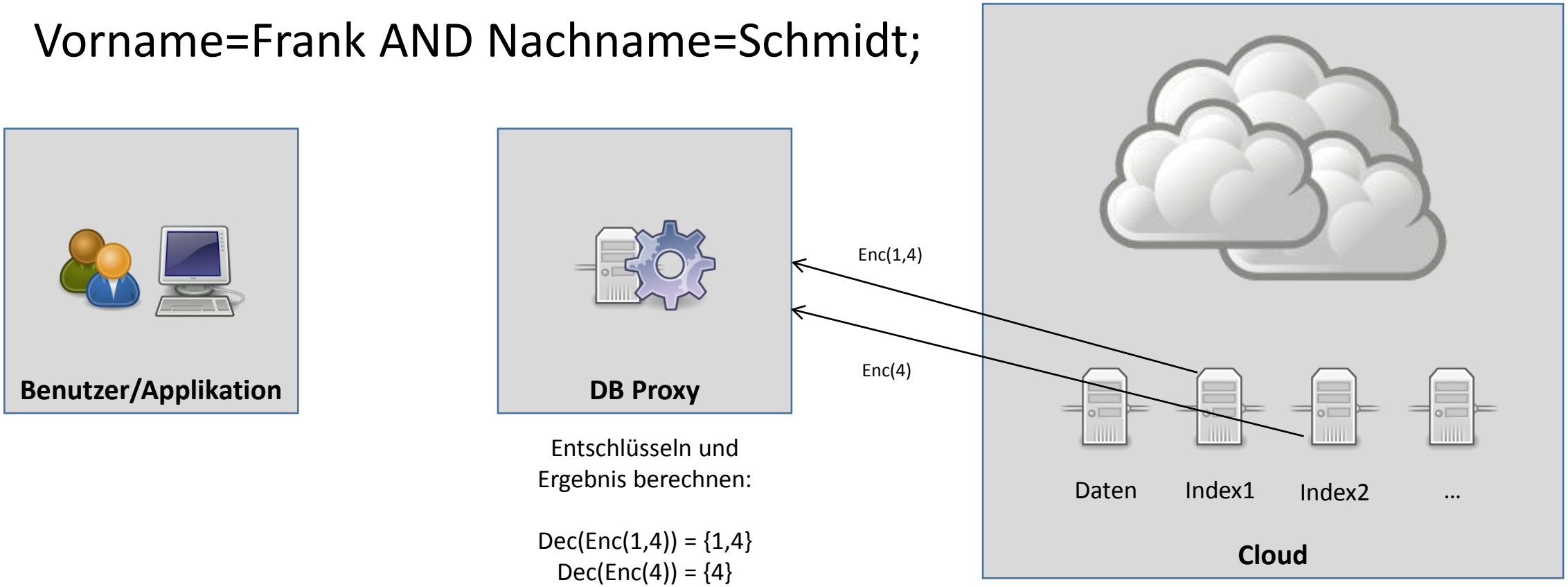
- SELECT * FROM customers WHERE Vorname=Frank AND Nachname=Schmidt;



Beispiel 2/4

ID	Vorname	Nachname	Geburtstag	Stadt
1	Frank	Maier	1.2.1980	Karlsruhe
2	Adam	Schulz	1.2.1990	Berlin
3	Hannes	Schneider	1.2.2000	Hamburg
4	Frank	Schmidt	1.2.2010	München

- SELECT * FROM customers WHERE Vorname=Frank AND Nachname=Schmidt;



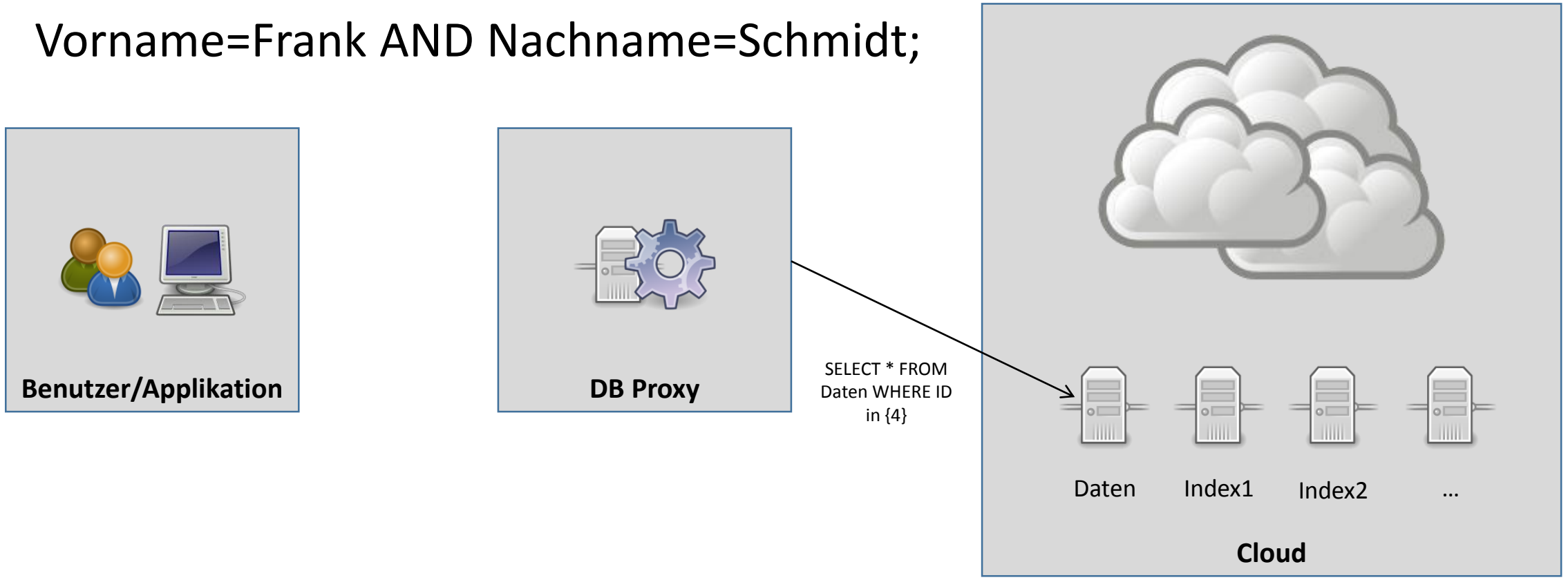
Entschlüsseln und Ergebnis berechnen:

Dec(Enc(1,4)) = {1,4}
Dec(Enc(4)) = {4}
{1,4} ∩ {4} = {4}

Beispiel 3/4

ID	Vorname	Nachname	Geburtstag	Stadt
1	Frank	Maier	1.2.1980	Karlsruhe
2	Adam	Schulz	1.2.1990	Berlin
3	Hannes	Schneider	1.2.2000	Hamburg
4	Frank	Schmidt	1.2.2010	München

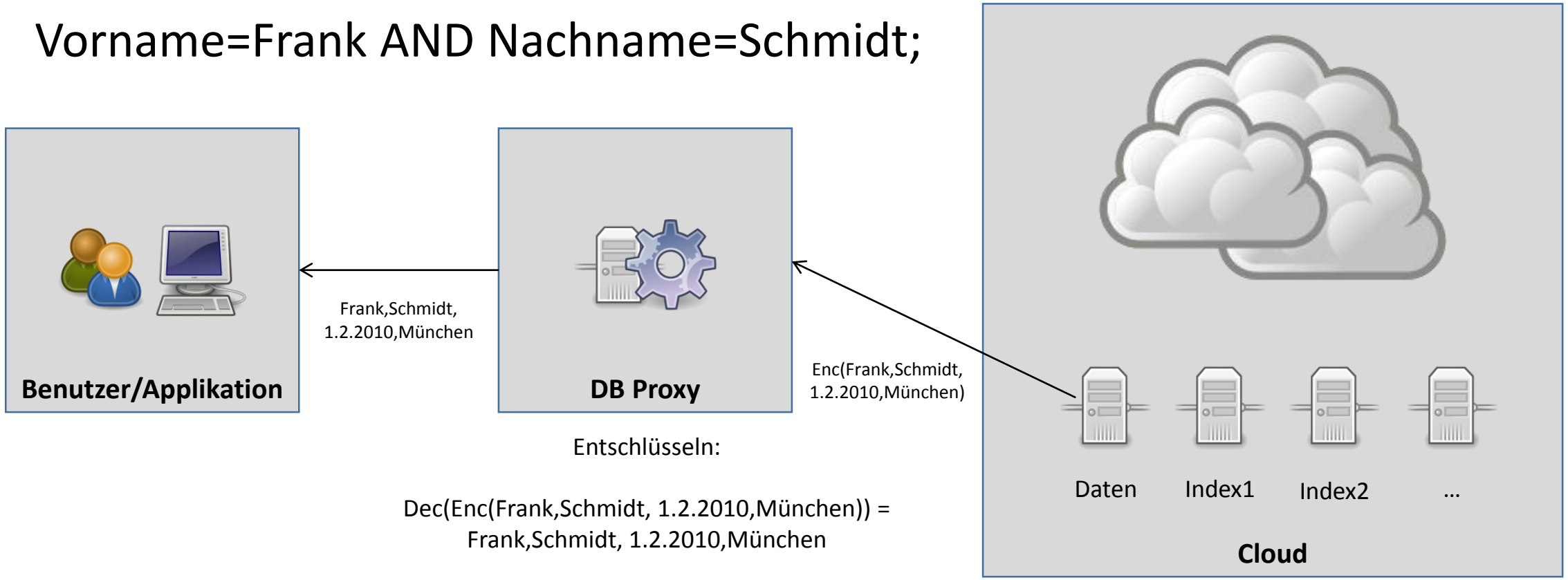
- SELECT * FROM customers WHERE Vorname=Frank AND Nachname=Schmidt;



Beispiel 4/4

ID	Vorname	Nachname	Geburtstag	Stadt
1	Frank	Maier	1.2.1980	Karlsruhe
2	Adam	Schulz	1.2.1990	Berlin
3	Hannes	Schneider	1.2.2000	Hamburg
4	Frank	Schmidt	1.2.2010	München

- SELECT * FROM customers WHERE Vorname=Frank AND Nachname=Schmidt;



Entschlüsseln:

Dec(Enc(Frank,Schmidt, 1.2.2010,München)) =
Frank,Schmidt, 1.2.2010,München

Praxisrelevante SQL-Befehle

- Bestimmung des relevanten SQL-Befehlssatzes aus Produktivsystemen

Datenbanksysteme

SIEMENS

UBITECH ubiquitous solutions

INTRASOFT INTERNATIONAL

CAS SOFTWARE

Extraktion relevanter SQL-Queries

SELECT COMMUNICATION_EVENT_ID, COMMUNICATION_EVENT_TYPE_ID, ORIG_COMM_EVENT_ID, PARENT_COMM_EVENT_ID, STATUS_ID, CONTACT_MECH_TYPE_ID, CONTACT_MECH_ID_FROM, PARTY_ID_TO, ENTRY_DATE, DATETIME_STARTED, DATETIME_ENDED, SUBJECT, CONTENT_MIME_TYPE_ID, CC_STRING, BCC_STRING, MESSAGE_ID, LAST_UPDATED_STAMP, LAST_UPDATED_TX_STAMP, CREATED_STAMP, CREATED_TX_STAMP **FROM** COMMUNICATION_EVENT **WHERE** ((COMMUNICATION_EVENT_TYPE_ID = ? **AND** STATUS_ID = ? **AND** CONTACT_LIST_ID **IS NOT NULL AND** (DATETIME_STARTED < ? **OR** DATETIME_STARTED **IS NULL**))) where clause:[COMMUNICATION_EVENT_TYPE_ID=COMMENT_NOTE, STATUS_ID=COM_IN_PROGRESS, DATETIME_STARTED=2015-11-23 11:01:21.401]

SELECT ORLE.ROLE_TYPE_ID, ORLE.PARTY_ID, COUNT(OH.ORDER_ID), **SUM**(OH.GRAND_TOTAL), **SUM**(OH.REMAINING_SUB_TOTAL) **FROM** ORDER_ROLE ORLE **INNER JOIN** ORDER_HEADER OH ON ORLE.ORDER_ID = OH.ORDER_ID **WHERE** ((ORLE.PARTY_ID = ? **AND** ORLE.ROLE_TYPE_ID = ? **AND** OH.ORDER_TYPE_ID = ? **AND** OH.STATUS_ID = ?)) **GROUP BY** ORLE.PARTY_ID, ORLE.ROLE_TYPE_ID where clause:[ORLE_PARTY_ID=admin, ORLE_ROLE_TYPE_ID=PLACING_CUSTOMER, OH_ORDER_TYPE_ID=SALES_ORDER, OH_STATUS_ID=ORDER_COMPLETED]

SELECT PAT.PY_ACTUAL_CURRENCY_UOM_ID **FROM** (**SELECT** PY.ACTUAL_CURRENCY_UOM_ID **AS** PY_ACTUAL_CURRENCY_UOM_ID **FROM** PAYMENT PY **INNER JOIN** PAYMENT_TYPE TY ON PY.PAYMENT_TYPE_ID = TY.PAYMENT_TYPE_ID) PAT **LEFT OUTER JOIN** CREDIT_CARD CC **ON** PAT.PY_PAYMENT_METHOD_ID = CC.PAYMENT_METHOD_ID **WHERE** ((PAT.PY_PAYMENT_METHOD_TYPE_ID = ?)) **ORDER BY** PAT.PY_EFFECTIVE_DATE DESC where clause:[PAT_PY_PAYMENT_METHOD_TYPE_ID=CASH]

…

Unterstützte SQL-Befehle

- Select, Update, Insert, Delete, Drop Table, Alter Table
- Joins
  - Left, Right, Outer, Inner, Full, …
- =, Not, In, And, Or
- Like (%)
- Subselect
- <, <=, >, >=
- Group by, Limit, Sum, Avg, Count

Leistungsfähigkeit

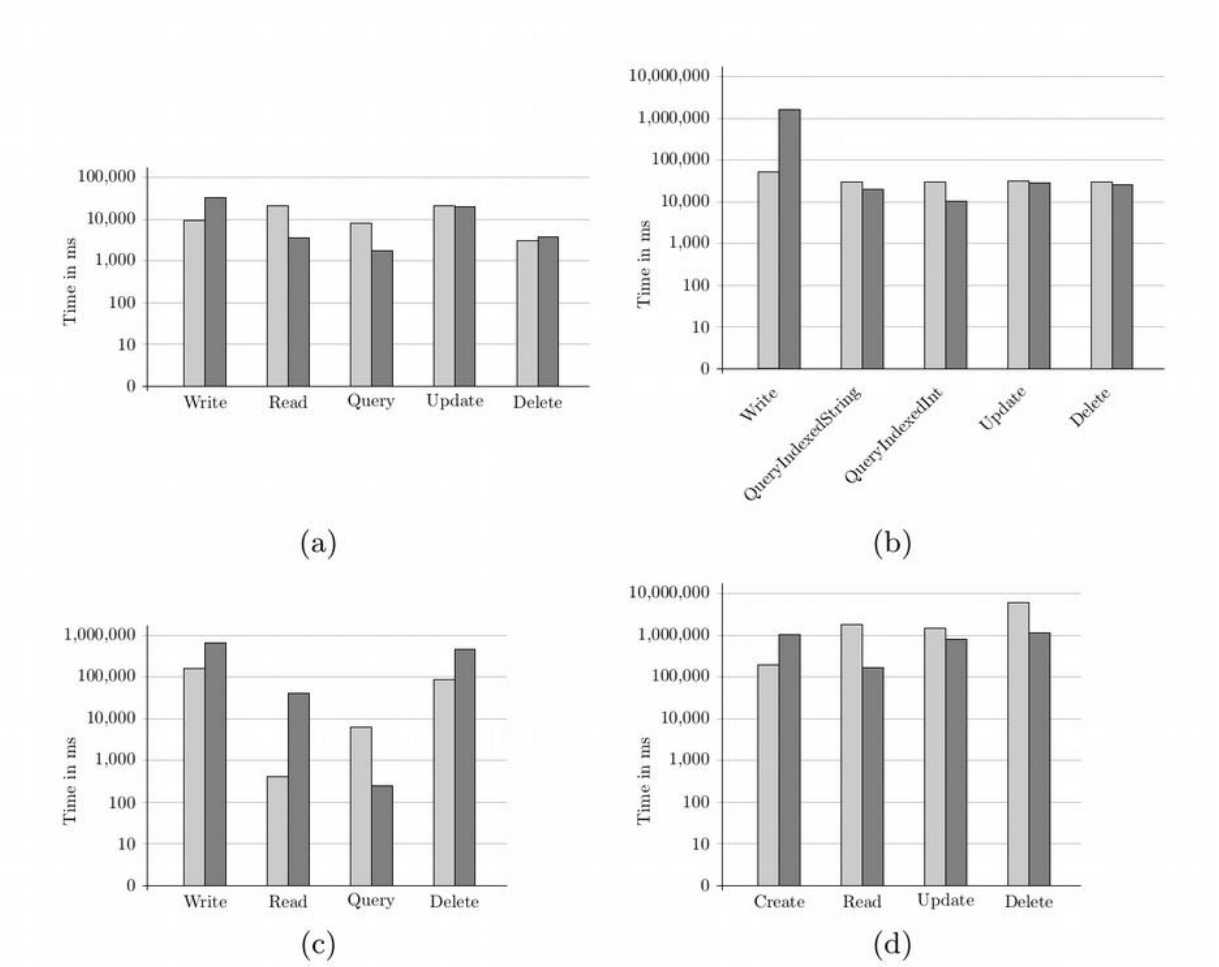


Fig. 11. Benchmark of circuit (a) *Complex*, (b) *Flatobject*, (c) *Inheritancehierarchy*, and (d) *Nestedlists* of JDO/DataNucleus without (light gray) and with the Cumulus4j plug-in (dark gray).

unsicher

abgesichert

Verbesserungen

- Sicherheit
  - Verschlüsselter Index
    - Deterministische Verschlüsselung
  - Verteilung
    - Fragmentierung und Speichern auf mehreren Servern
  - Trusted Execution
    - Intel SGX
- Leistungsfähigkeit
- Funktionalität
  - SQL-Unterstützung

Transformation mit verschlüsseltem Index

ID	Vorname	Nachname	Geburtstag	Stadt
1	Frank	Maier	1.2.1980	Karlsruhe
2	Adam	Schulz	1.2.1990	Berlin
3	Hannes	Schneider	1.2.2000	Hamburg
4	Frank	Schmidt	1.2.2010	München

Original

ID	Encrypted Data
1	Enc(Frank, Maier, 1.2.1980, Karlsruhe)
2	Enc(Adam, Schulz, 1.2.1990, Berlin)
3	Enc(Hannes, Schneider, 1.2.2000, Hamburg)
4	Enc(Frank, Schmidt, 1.2.2010, München)

Daten

Keyword	IDs
Enc(Vorname:Frank)	Enc(1, 4)
Enc(Vorname:Adam)	Enc(2)
Enc(Vorname:Hannes)	Enc(3)

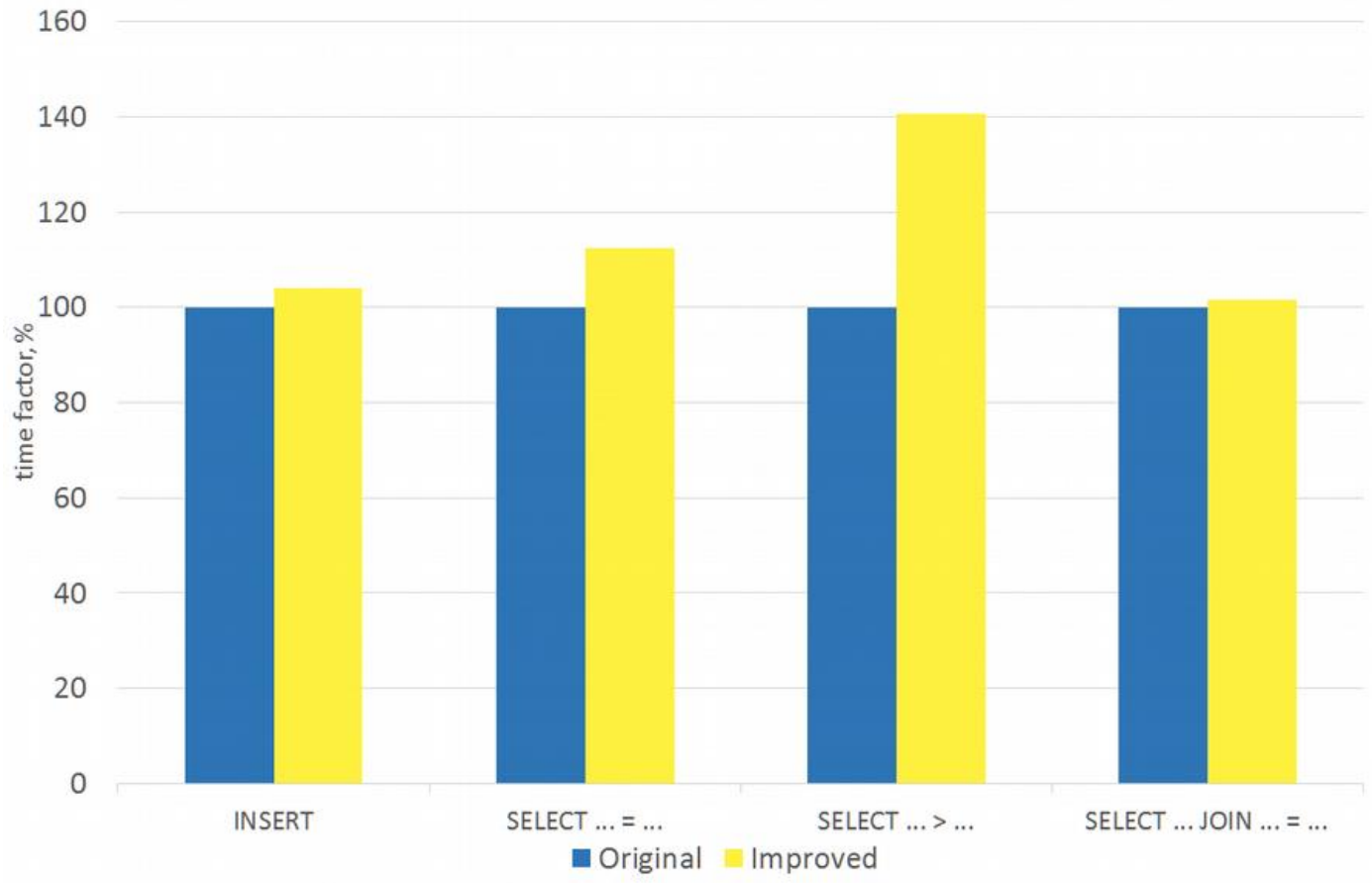
Index 1

Keyword	IDs
Enc(Nachname:Maier)	Enc(1)
Enc(Nachname:Schulz)	Enc(2)
Enc(Nachname:Schneider)	Enc(3)
Enc(Nachname:Schmidt)	Enc(4)

Index 2

Enc(Original)

Leistungsfähigkeit mit verschlüsseltem Index



Transformation mit Datenverteilung

ID	Encrypted Data
1	Enc(Frank, Maier, 1.2.1980, Karlsruhe)
2	Enc(Adam, Schulz, 1.2.1990, Berlin)
3	Enc(Hannes, Schneider, 1.2.2000, Hamburg)
4	Enc(Frank, Schmidt, 1.2.2010, München)

Daten

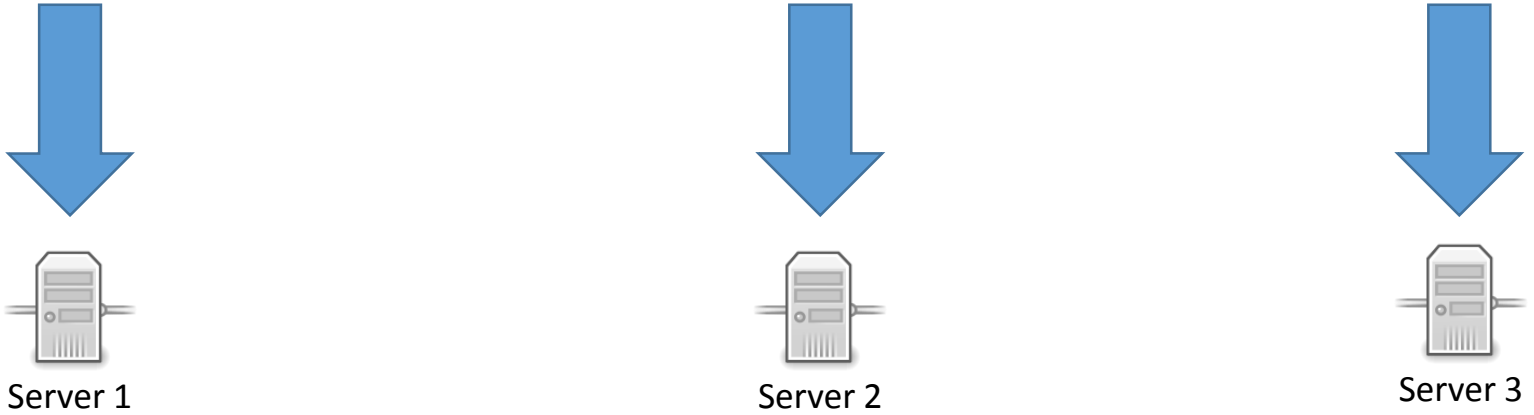
Keyword	IDs
Vorname:Frank	Enc(1, 4)
Vorname:Adam	Enc(2)
Vorname:Hannes	Enc(3)

Index 1

Keyword	IDs
Nachname:Maier	Enc(1)
Nachname:Schulz	Enc(2)
Nachname:Schneider	Enc(3)
Nachname:Schmidt	Enc(4)

Index 2

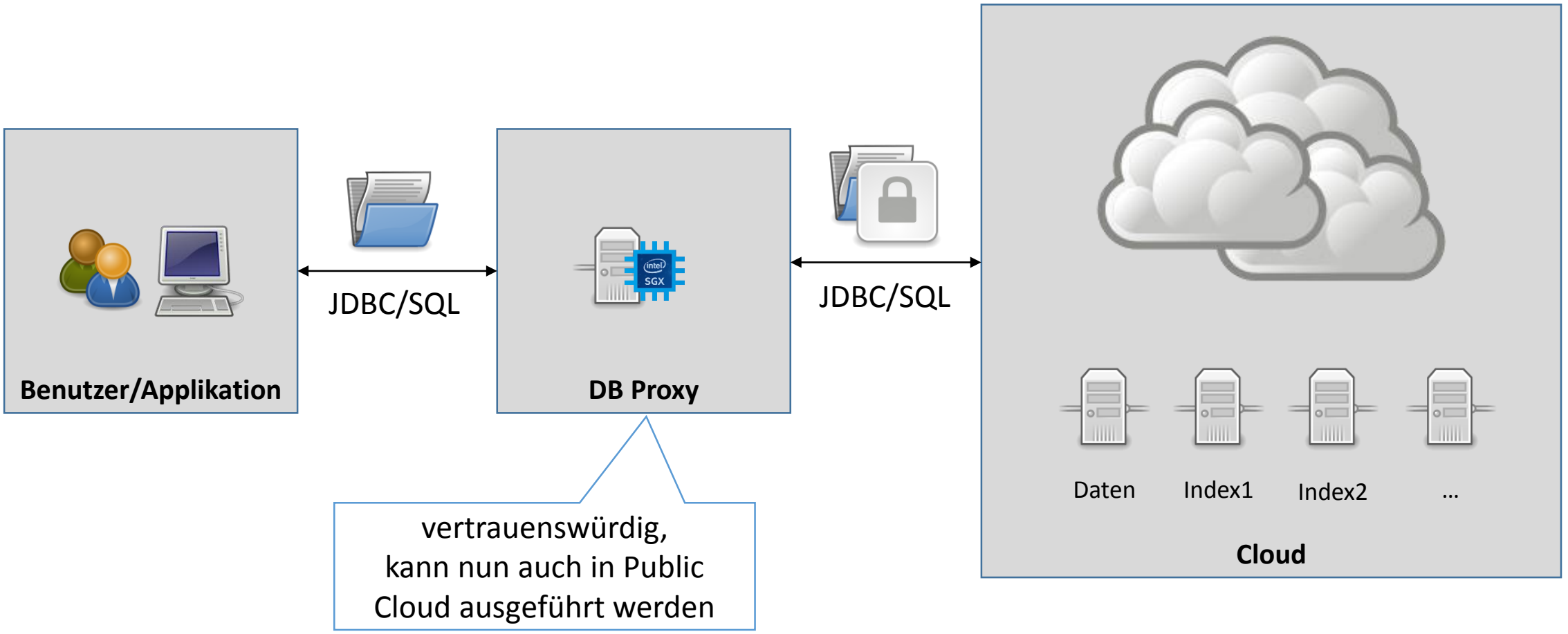
Enc(Original)



Intel Software Guard Extensions (SGX)

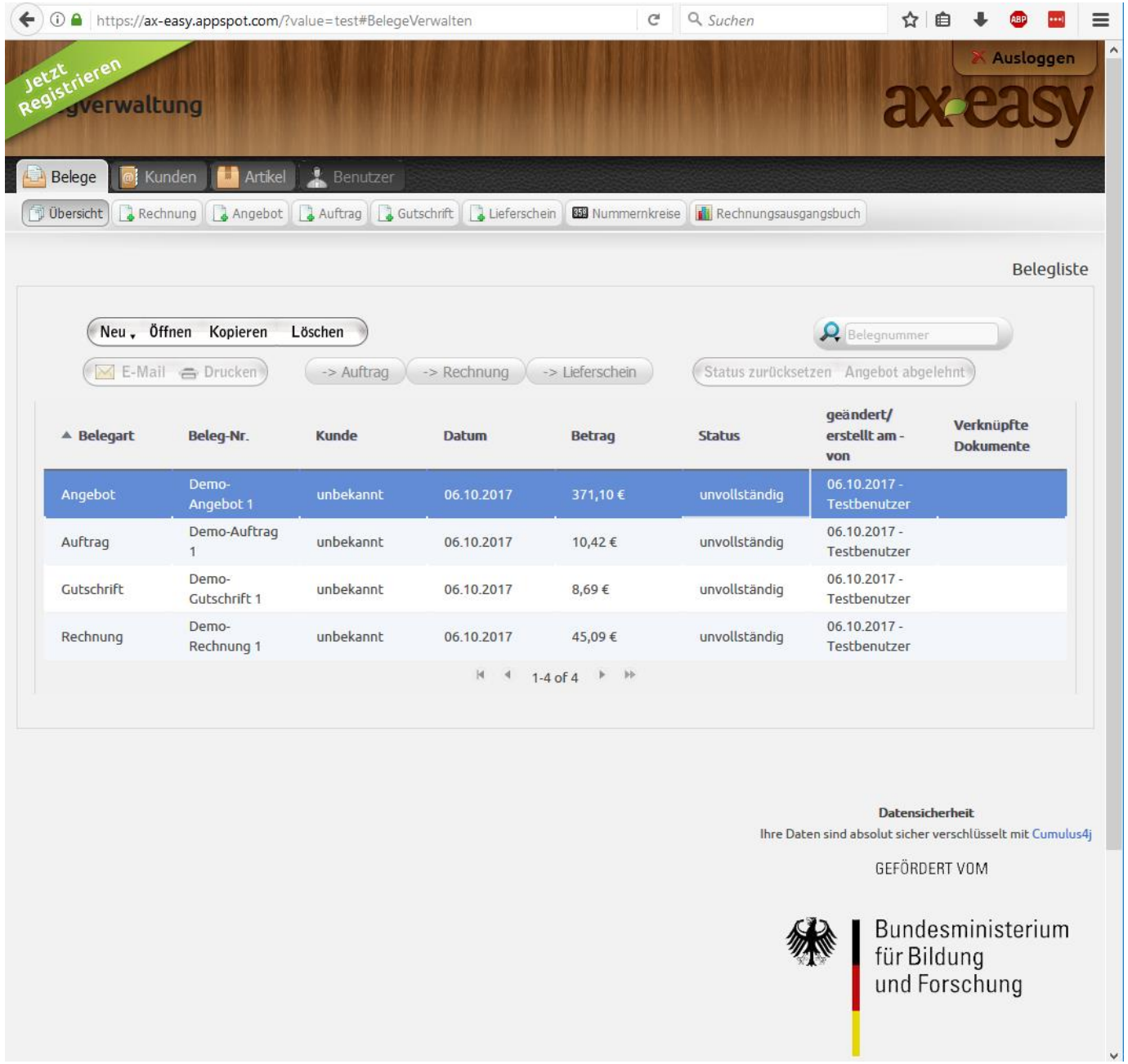
- Vorteile
  - Isolation, Attestierung, sichere Schlüsselspeicherung
  - Grundidee: Selbst privilegierte Prozesse können geschützte Speicherbereiche nicht ansprechen
  - Daher: Sichere Ausführung von Programmen selbst in kompromittierten Umgebungen
- Nachteile
  - Schlechtere Performance (eingeschränkter Speicher)
  - Vertrauen in Intel notwendig
  - Geschützte Bereiche sind nicht mehr überwachbar
  - Erste Angriffe („Malware Guard Extension: Using SGX to Conceal Cache Attacks“, Schwarz et al., 2017)

Architektur mit SGX



Demonstratoren

- ax-easy
  - kommerzielles Produkt
  - basierend auf Cumulus4j
  - www.ax-easy.de



Demonstratoren (CEBIT 2011-2014)

Zusammenfassung

- Zwischenschicht verschlüsselt sensible Daten
- Datenkontrolle bleibt beim Anwender
- Schlüssel kommt nie in die Cloud
- Praktisch einsetzbar

Eigene Beiträge

- Verfahren [7] und Sicherheitsbeweis [6]
- Implementierungen [3,5,6]
- Seitenkanäle [1,3]
- Verbesserungen der Sicherheit [1,3]
- Verbesserte Funktionalität [1,3]
- Verbesserte Leistungsfähigkeit [1,3,6]
- Verteilte Schlüsselverwaltung [2,4]

Eigene Publikationen

[1] Matthias Gabel, Jeremias Mechler
Secure Database Outsourcing to the Cloud: Side-Channels, Counter-Measures and Trusted Execution with SGX
30th IEEE International Symposium on Computer-Based Medical Systems (CBMS) 2017

[2] Gunther Schiefer, Murat Citak, Andreas Schoknecht, Matthias Gabel, Jeremias Mechler
Security in a Distributed Key Management Approach
30th IEEE International Symposium on Computer-Based Medical Systems (CBMS) 2017

[3] Rafael Dowsley, Matthias Gabel, Kateryna Yurchenko, Valentin Zipf
A Database Adapter for Secure Outsourcing
8th IEEE International Conference on Cloud Computing Technology and Science (CloudCom) 2016

[4] Rafael Dowsley, Matthias Gabel, Gerald Hübsch, Gunther Schiefer, Antonia Schwichtenberg
A Distributed Key Management Approach
8th IEEE International Conference on Cloud Computing Technology and Science (CloudCom) 2016

[5] Matthias Gabel, Gerald Hübsch
Secure Database Outsourcing to the Cloud Using the MimoSecco Middleware
Trusted Cloud Computing 2014

[6] Matthias Huber, Matthias Gabel, Marco Schulze, Alexander Bieber
Cumulus4j: A Provably Secure Database Abstraction Layer
ARES Conference: The Second International Workshop on Modern Cryptography and Security Engineering (MoCrySen) 2013

[7] Dirk Achenbach, Matthias Gabel, Matthias Huber
MimoSecco: A Middleware for Secure Cloud Storage
18th ISPE International Conference on Concurrent Engineering (CE 2011) / Improving Complex Systems Today

E. Papatheocharous, M. Abdelraheem, T. Carnehult, P. Gouvas, G. Schiefer, S. Mantzouratos, J. Mechler, M. Gabel, K. Yurchenko, S. Schork, G. Moldovan
PaaSword: A Data Privacy and Context-aware Security Framework for the Cloud
to be submitted (2018)

Günter Saur, Stéphane Estable, Karin Zielinski, Stefan Knabe, Michael Teutsch, Matthias Gabel
Detection and Classification of man-made Offshore Objects in TerraSAR-X and RapidEye Imagery
Proceedings of IEEE Oceans, 2011

Forschungsgruppen

- Italien (Università degli Studi di Milano)
  - 2007-2011 Samarati, Ciriani: Fragmentierung (+ Verschlüsselung)
- Karlsruhe (KIT)
  - 2010-2017 Huber, Gabel: Cumulus4j/PaaSword
  - 2011-2016 Köhler (ITM Hartenstein): Privacy Constraints
  - Schaad/Kerschbaum: SEEED (SAP SE): Data Processing on SAP HANA (basierend auf CryptDB)
- USA (MIT CSAIL)
  - 2011-2014 Zeldovich, Popa: CryptDB -> PreVeil

Zitierte Publikationen

[Aggarwal2005] Gagan Aggarwal, Mayank Bawa, Prasanna Ganesan, Hector Garcia-Molina, Krishnaram Kenthapadi, Rajeev Motwani, Utkarsh Srivastava, Dilys Thomas, and Ying Xu
Two Can Keep a Secret: A Distributed Architecture for Secure Database Services
Second Biennial Conference on Innovative Data Systems Research (CIDR 2005), 2005

[Gentry2009] Craig Gentry
A Fully Homomorphic Encryption Scheme
PhD Thesis, Stanford University, 2009

[Gentry2015] Craig Gentry, Shai Halevi, Nigel P. Smart
Homomorphic Evaluation of the AES Circuit (Updated Implementation)
Cryptology ePrint Archive: Report 2012/099, Extended abstract in CRYPTO 2012

[Goldreich1987] Oded Goldreich, Silvio Micali, Ari Widgerson
How to Play Any Mental Game
Symposium on Theory of Computing, STOC ’87, New York, 1987

[Hacigümüs2002] Hakan Hacigümüş, Bala Iyer, Chen Li, and Sharad Mehrotra
Executing sql over encrypted data in the database-service-provider model
ACM SIGMOD International Conference on Management of Data, SIGMOD ’02, pages 216–227, New York, 2002

[Popa2011] Raluca Ada Popa, Nickolai Zeldovich, and Hari Balakrishnan
CryptDB: A Practical Encrypted Relational DBMS
Technical Report MIT-CSAIL-TR-2011-005, Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, 2011

Verwendete Grafiken

- Enigma
  ITI, KIT
- Server im Rechenzentrum
  SDIL, KIT
- Schnecke
  http://cheezburger.com/1005160704
- Penrose-Dreieck
  - https://de.wikipedia.org/wiki/Paradoxon#/media/File:Penrose-dreieck.svg
- CryptDB
  - http://people.csail.mit.edu/nickolai/papers/raluca-cryptdb-tr.pdf
- Two Can Keep a Secret (Aggarwal 2005)
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.9485&rep=rep1&type=pdf
- Executing SQL Over Encrypted Data... (Hacigümüs 2002)
  - http://www.ics.uci.edu/~chenli/pub/sigmod02.pdf
- PRISM
  - https://commons.wikimedia.org/wiki/File:PRISM_Collection_Details.jpg (abgerufen am 27.03.2017)
- WikiLeaks
  - https://www.wikileaks.org/ (abgerufen am 06.10.2017)
- RAID
  - https://de.wikipedia.org/wiki/RAID
- Sonstige Symbole
  https://commons.wikimedia.org/wiki/Tango_icons

Verwendete Logos

- Gmail
  - https://play.google.com/store/apps/details?id=com.google.android.gm&hl=de
- Office 365
  - https://de.wikipedia.org/wiki/Microsoft_Office_365#/media/File:Office_365_logo.png
- Outlook.com
  - https://www.microsoft.com/de-de/outlook-com/img/Outlook_Facebook_Icon.jpg
- Dropbox
  - https://cfl.dropboxstatic.com/static/images/logo_catalog/logotype_2016-vfliXTM4h.svg
- Amazon Web Services
  - https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/AmazonWebservices_Logo.svg/1200px-AmazonWebservices_Logo.svg.png
- SQL Azure
  - https://blogs.technet.microsoft.com/dataplatforminsider/2009/07/08/microsoft-sql-services-is-now-microsoft-sql-azure/
- Google Compute Engine
  - https://upload.wikimedia.org/wikipedia/en/1/18/Google_Compute_Engine_logo.png
- Siemens
  - https://de.wikipedia.org/wiki/Siemens#/media/File:Siemens-logo.svg
- Ubitech
  - https://www.ubitech.eu/wp-content/uploads/logo.png
- Intrasoft
  - http://www.jobfairathens.gr/system/companies/logos/210/Intrasoft-logo-trans.png?1459804136
- CAS
  - http://www.cas.de/fileadmin/layout/images/cas-logo.png
- Wikileaks
  - https://commons.wikimedia.org/wiki/File:Wikileaks_logo.svg
- …
  - .

Sicherheitsbeweis

- Annahmen
  - Verschlüsselung ist IND-CPA-sicher
  - Kein Hintergrundwissen
  - Chiffratlängen identisch etc.
- Indirekter Beweis
  - Angreifer könnte IND-ICP brechen
  - Dann muss er schon IND-CPA brechen können
- Widerspruch zur Annahme
- Formal bewiesen in [6]

RAID

- RAID = Redundant Array of Independent Disks
- RAID 0: Striping (reine Verteilung ohne Redundanz)
- RAID 1: Mirroring (einfache Redundanz)
- RAID 5: Block-Bevel Striping + Distributed Parity