DEUTSCHE INITIATIVE FÜR NETZWERKINFORMATION E.V.

Technische Aspekte des DINI-Zertifikats 2007

Tutorial: Aspekte der Authentizität und Integrität der Dokumente

Dr. Judith Plümer, Thomas Severiens

judith@mathematik.uni-osnabrueck.de

severiens@mathematik.uni-osnabrueck.de

Arbeitsgruppe Mathematische Fachinformation

Fachbereich Mathematik/Informatik

Universität Osnabrück

Inhaltsüberblick

- Motivation
- Fehler-Szenarien und daraus resultierende Strategien und Techniken
  - SSL / TLS
  - Zertifikate
  - Hashes
  - Migration
- Integration der Anforderungen in die Repository Infrastruktur
- Empfehlungen

Motivation

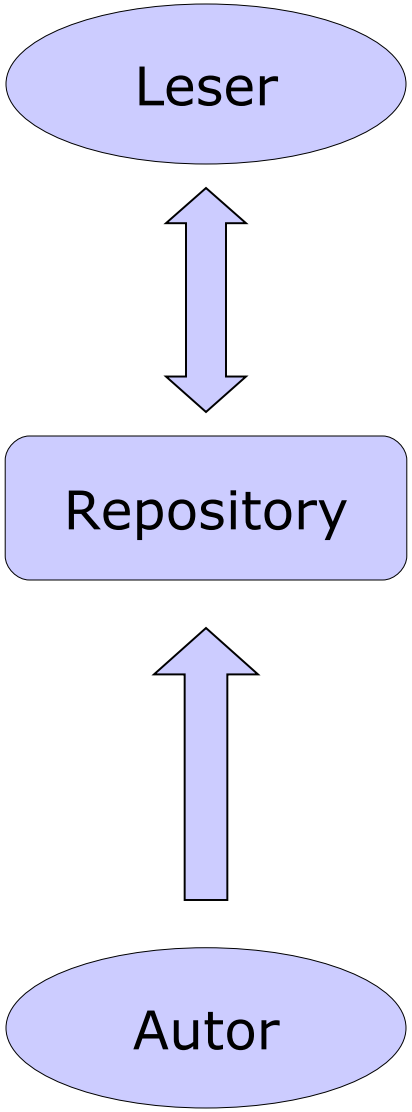
Empfehlungen aus dem DINI-Zertifikat 2007 (Auszug):

- 2.5.1: SSL-*Zertifizierung* mit vertrauenswürdigem Zertifikat für verschlüsselte Kommunikation wird eingesetzt.
- 2.5.2: Einsatz eines Verfahrens zum Nachweis der Unversehrtheit der Dokumente (Hash-Wert) sowie Veröffentlichung von Verfahren und Hash-Werten.
- 2.5.2: Fortgeschrittene digitale Signatur wird verwendet.

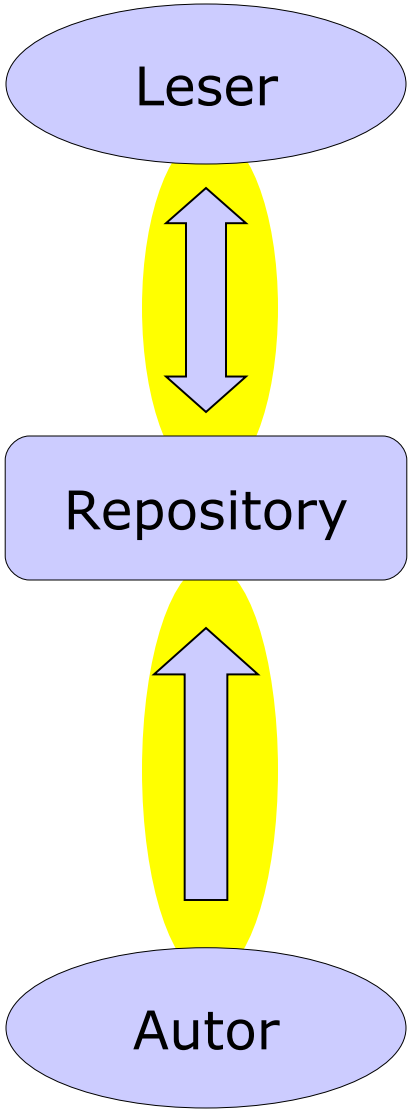
Ausgangs-Szenarium

- Autor legt Publikation in Repository
- Leser recherchiert im Repositoy
- Kommunikation offen

- Viele Arten von Manipulation möglich



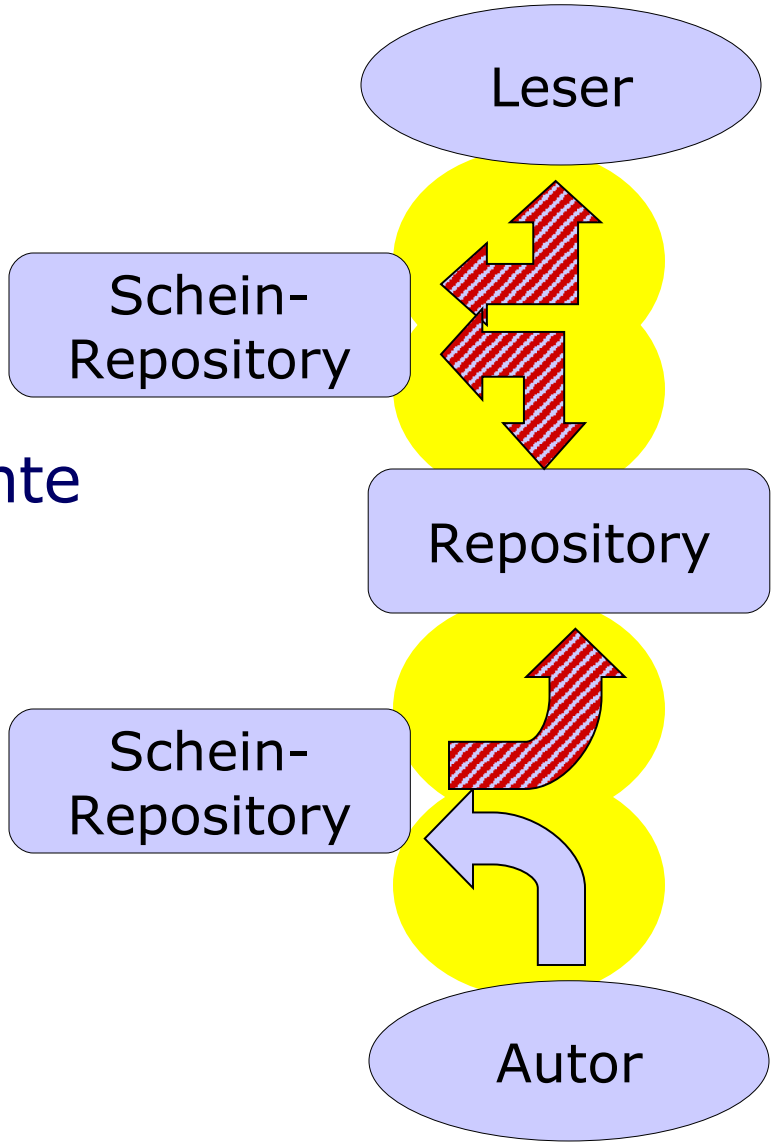
Abhören von Login-Daten



- Angriff: Abhören von Passworten, Manipulation auf der Leitung, ...
- Lösung: Verschlüsselte Leitung (SSL/TLS)

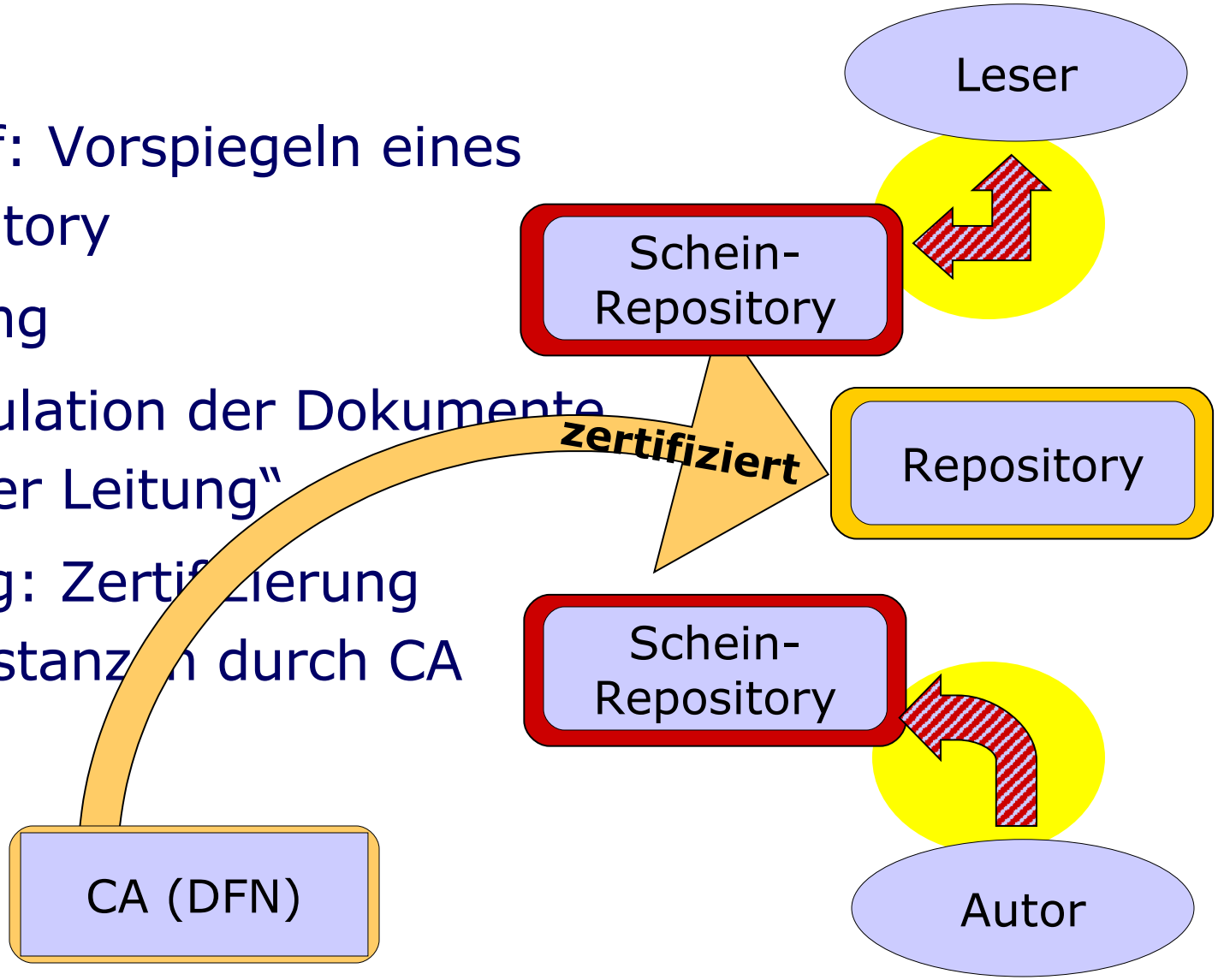
„Man in the middle“

- Angriff: Vorspiegeln eines Repository
- Phishing
- Manipulation der Dokumente „auf der Leitung“
- Lösung: Zertifizierung der Instanzen durch CA



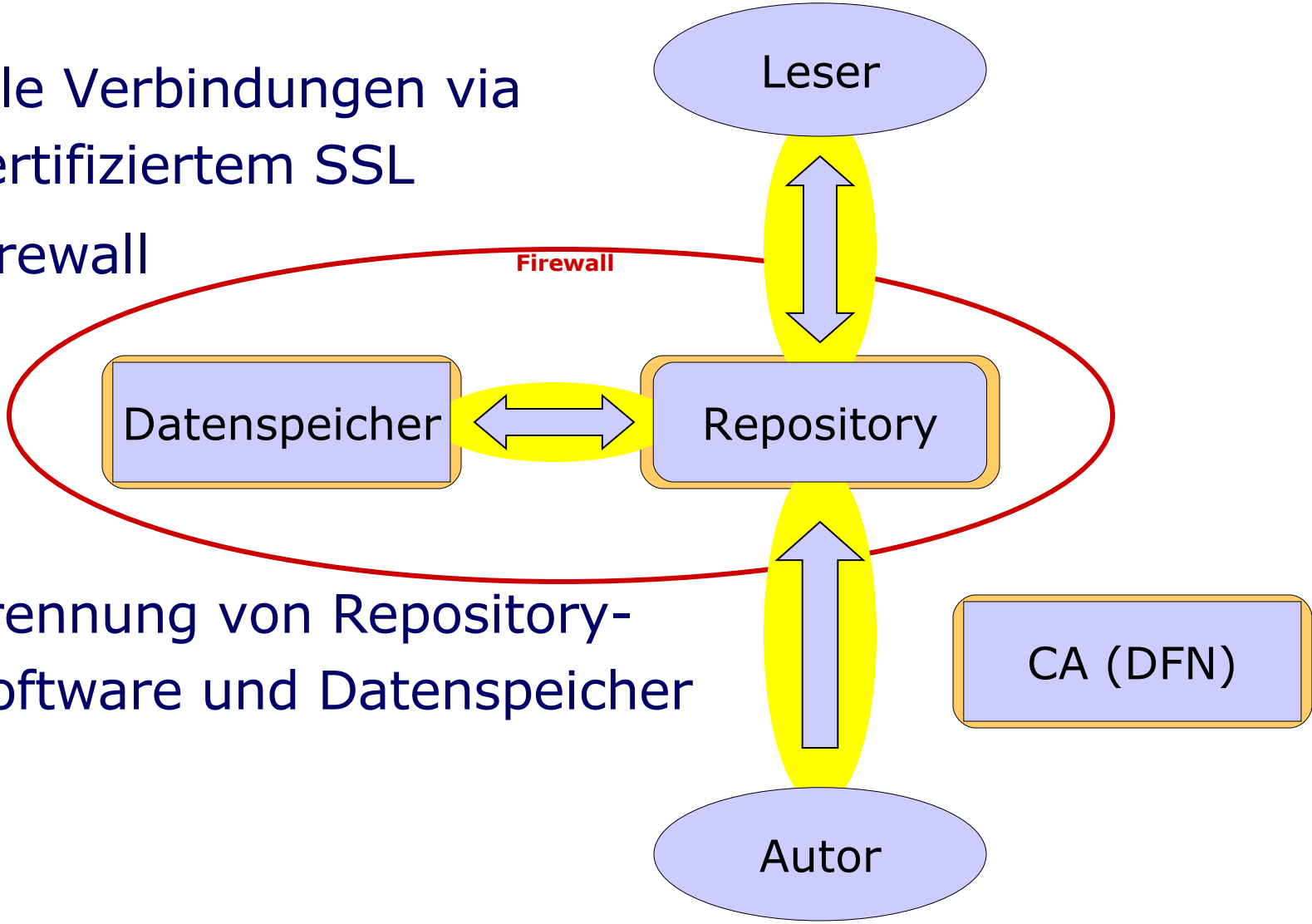
„Man in the middle“

- Angriff: Vorspiegeln eines Repository
- Phishing
- Manipulation der Dokumente „auf der Leitung“
- Lösung: Zertifizierung der Instanzen durch CA



Repository kommuniziert nur über zertifizierte SSL-Verbindungen

- Alle Verbindungen via zertifiziertem SSL
- Firewall



- Trennung von Repository-Software und Datenspeicher

Exkurs 1:

SSL
Secure Socket Layer

TLS
Transport Layer Security

Exkurs 0,5:

Verschlüsselungsverfahren

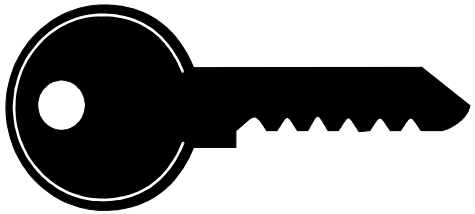
Verschlüsselungsverfahren

Geheim-Schlüsselverfahren

Secret-Key-Verfahren

Symmetrische Verfahren

„wer verschlüsseln kann, kann auch entschlüsseln"

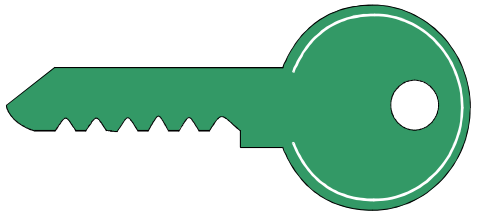


Öffentliche-Schlüssel-

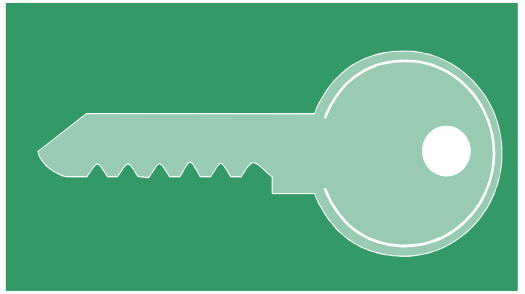
Public-Key-

Asymmetrische Verfahren

„ein Schlüssel zum kodieren,



ein Schlüssel zum dekodieren"



Verschlüsselungsverfahren

Secret-Key-Verfahren:

z.B. DES (Digital Encryption Standard)

+ Schnell zu berechnen

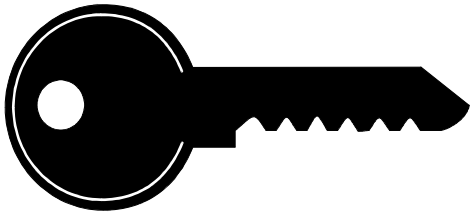
– Transport des Schlüssels ist unsicher

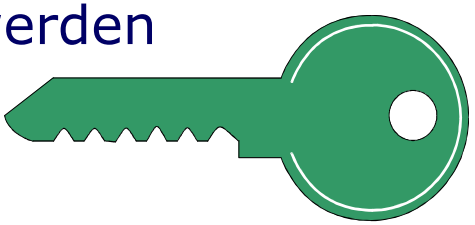
Public-Key-Verfahren

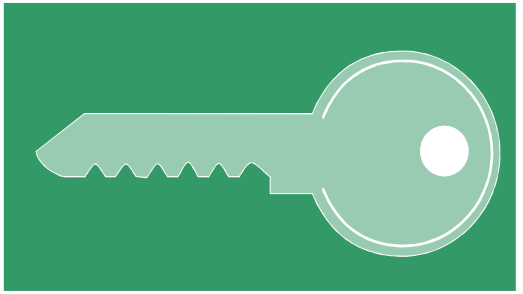
z.B. RSA (Rivest, Shamir, Adelman)

– Aufwendige Rechnungen

+ der geheime Schlüssel muss nicht transportiert werden



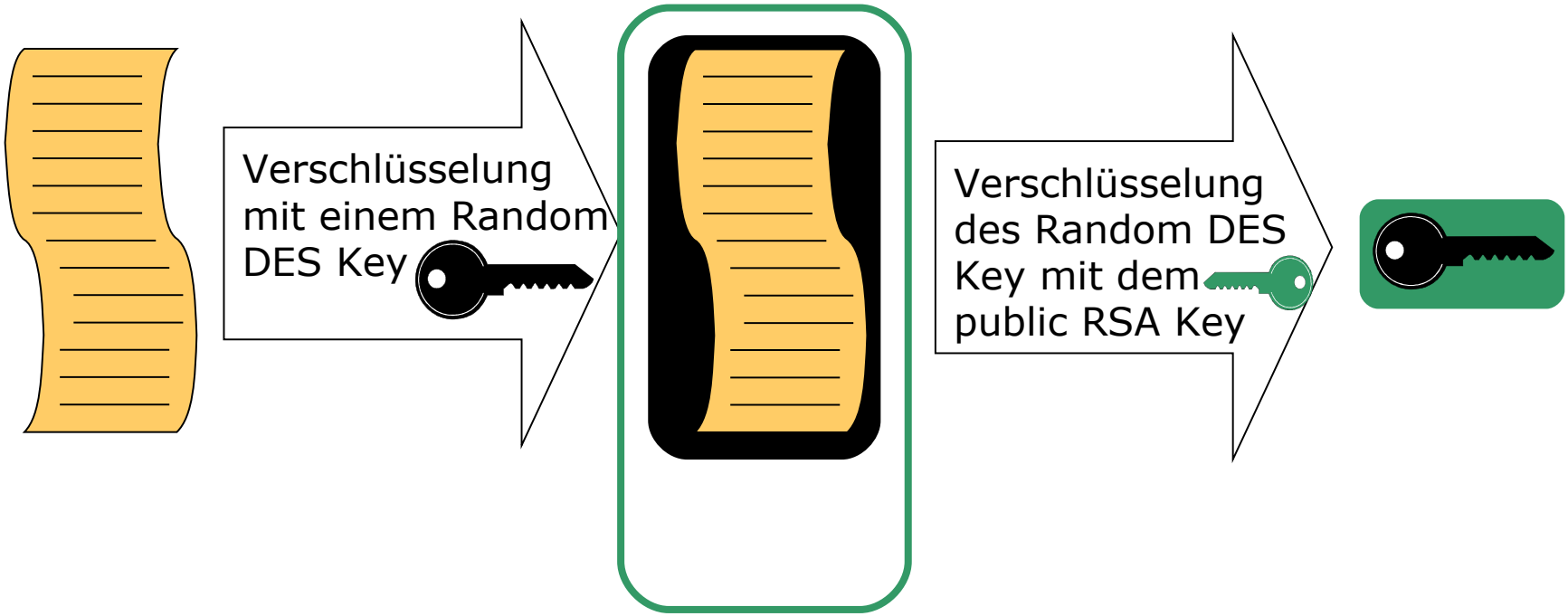




Verschlüsselungsverfahren

In der Praxis werden Hybrid-Verfahren genutzt:

Beispiel: Verschlüsselung eines Textes mit der PGP Software (Pretty Good Privacy)



Exkurs 1:

SSL
Secure Socket Layer

TLS
Transport Layer Security

TLS Transport Layer Security 1.0/1

- Basiert auf SSL (Secure Socket Layer) 3.0 (entwickelt von Netscape, 1996) allerdings nicht kompatibel
- RFC2246 (1999), RFC4346(2006)
- Ziel:
  - Privacy und Data Integrity zwischen zwei kommunizierenden Anwendungen (z.B. Server und Browser)
- Idee:
  - Symmetrische kryptographische Verfahren (DES) werden benutzt, der jeweilige Schlüssel wird für jede Verbindung neu generiert

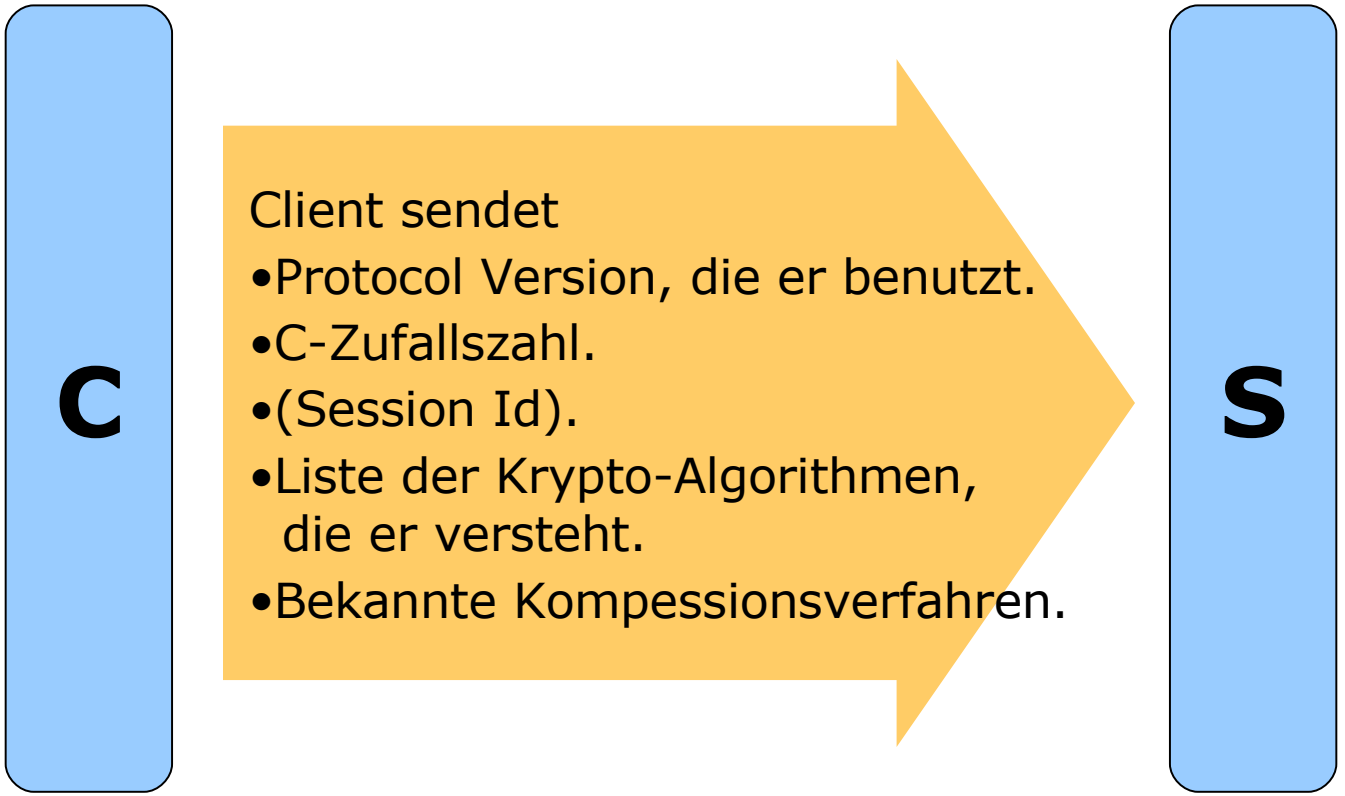
TLS

- Besteht aus zwei Protokollen:
- Handshake protocol
  - Public-Key Verfahren werden genutzt um einen gemeinsamen Secret Key zu generieren
- Record protocol
  - Der gemeinsame Secret Key wird nun zum Schutz der Kommunikation zwischen zwei Parteien (z.B. Client, Server) genutzt. D.h. die Nachrichten werden mit dem Secret Key verschlüsselt.
- Wir interessieren uns im Folgenden für das Handshake protocol

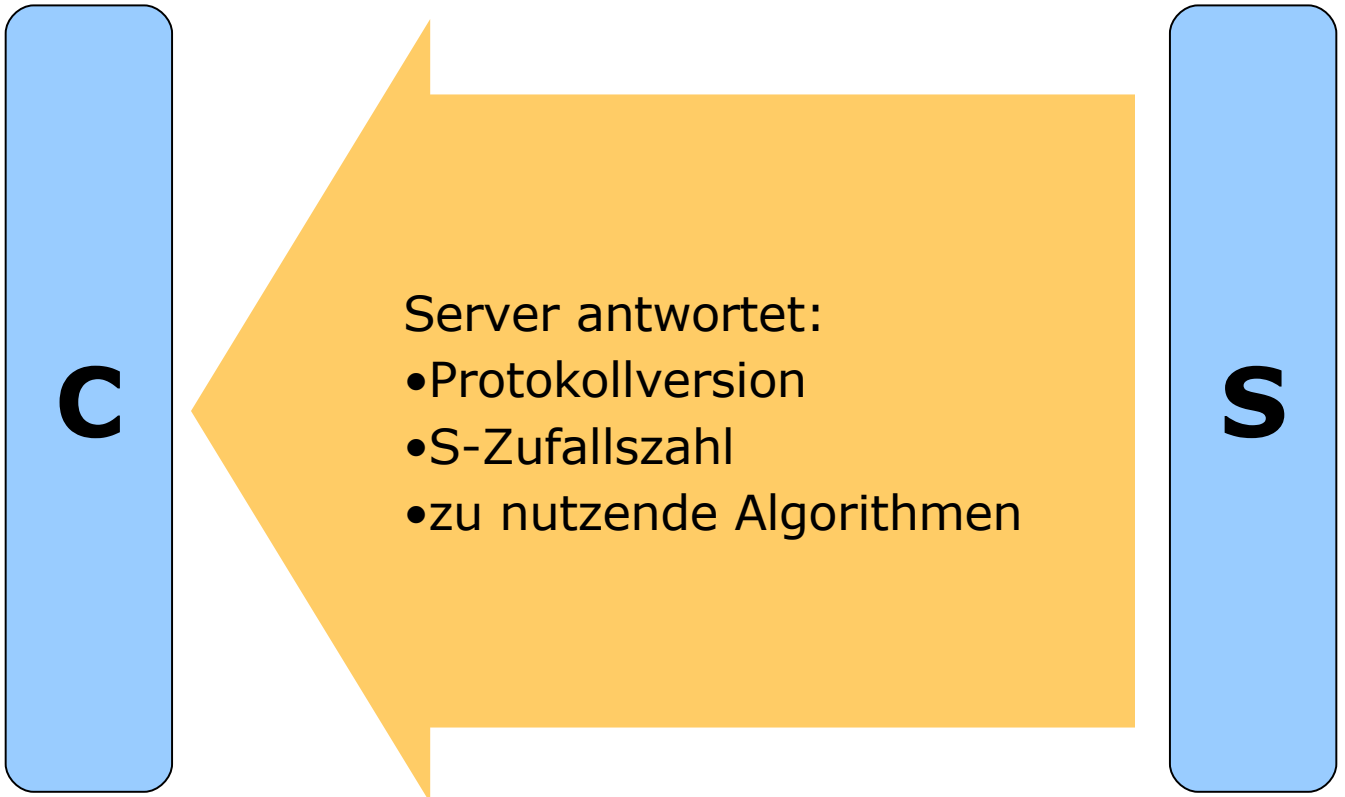
TLS Handshake Protocol

- Vereinbarung über das benutzte Protokoll und die zu nutzenden (Krypto)Algorithmen
  - Interoperabilität
- Authentifizierung von Client und Server (optional)
  - Mitteilung des Public Key und Überprüfung des Zertifikates
- Public Key wird genutzt um einen gemeinsamen Secret Key zu erzeugen

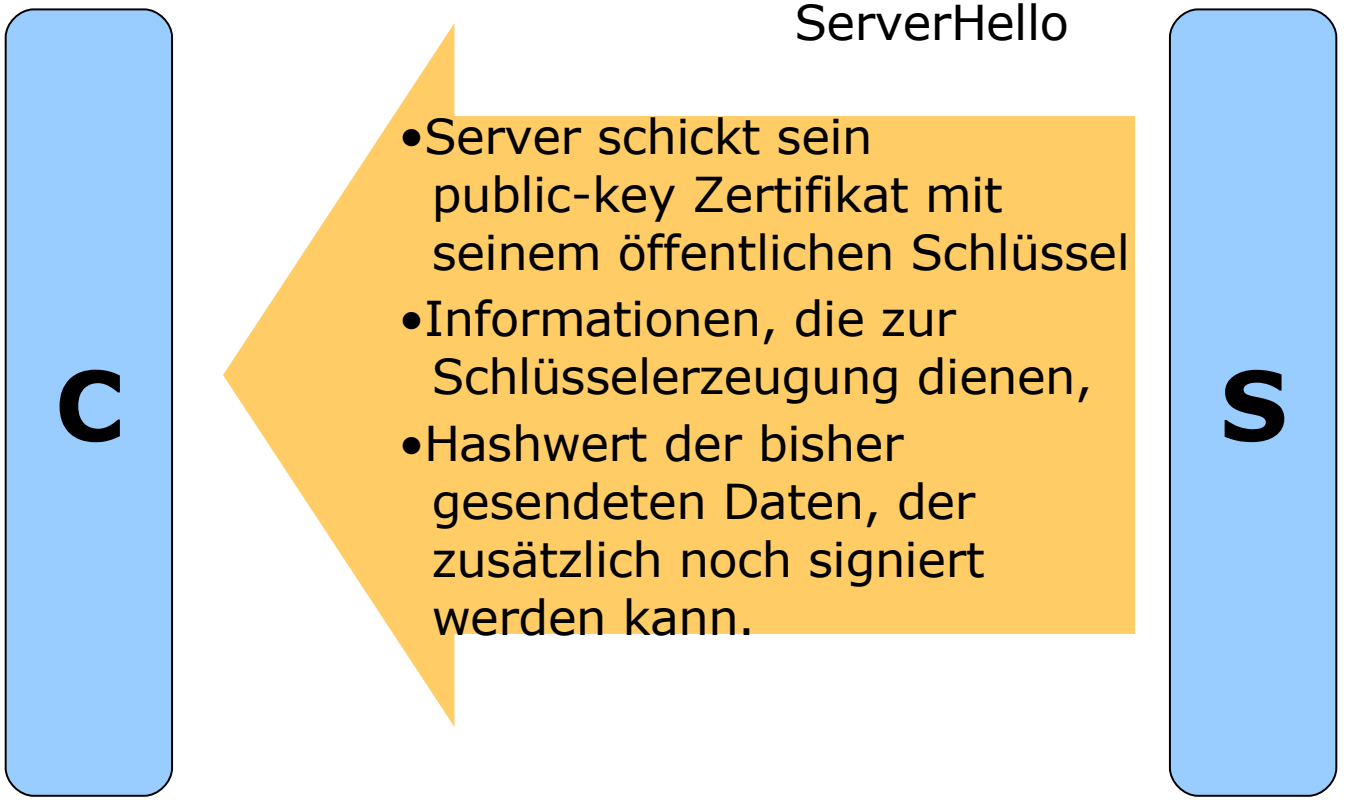
ClientHello



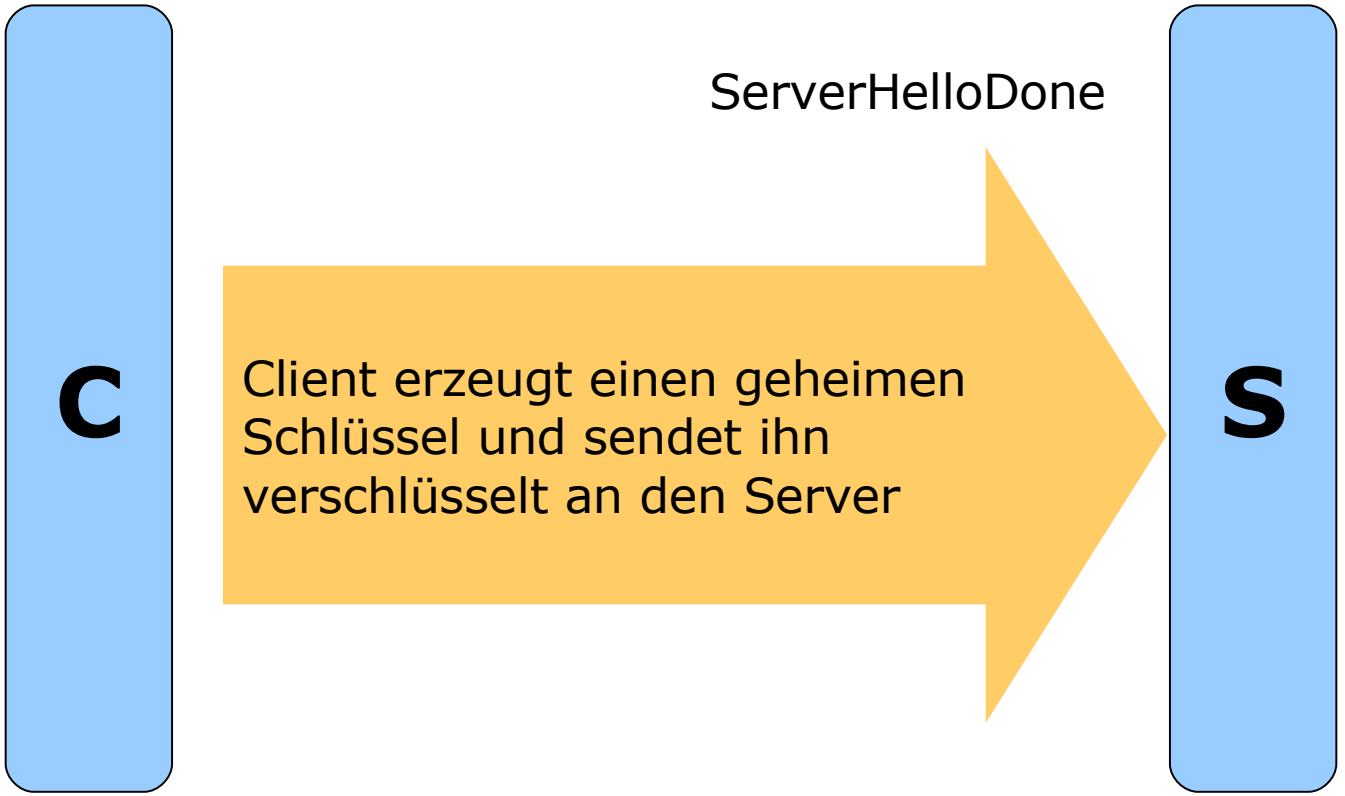
ServerHello



ServerKeyExchange



ClientKeyExchange



TLS – konkrete Anwendung

- Generierung eines gemeinsamen Schlüssels, wenn beide Parteien über zertifizierte Public Keys verfügen.
- Generierung eines gemeinsamen Schlüssels, wenn nur der Server über zertifizierte Public Keys verfügt (keine echte Sicherheit für den Server).
- Generierung eines gemeinsamen Schlüssels ohne Zertifikate ist möglich, allerdings anfällig für Man-in-the-middle-attacks.

Exkurs 2

Zertifikate

Exkurs 2: Zertifikate: Rechtliche Einbettung

- Rechtliche Regelung: Signaturgesetz (SigG) aus 2001 („Gesetz über Rahmenbedingungen für elektronische Signaturen...“)
- Regelt nur **qualifizierte Zertifikate**, also solche, die durch einen Zertifizierungsdiensteanbieter ausgestellt wurden.
- Aufsichtsbehörde: Bundesnetzagentur
- Strenge Anforderungen an das Sicherheitskonzept (also gibt es kaum Zertifikatsanbieter)...

Exkurs 2: Zertifikate

- Zertifizierungsstelle muss Zertifikate jederzeit sperren können (§5 SigG)
- Archivierungspflicht für wenigstens 30 Jahre. (SigO)
- Akkreditierung des Zertifizierungsdiensteanbieters möglich (§15 SigG), wenn alle Forderungen eingehalten werden. Damit ist der Übergang vom **qualitifizierten** zum **akkreditierten Zertifikat** bestimmt.
- Nur akkreditierte Zertifikate bieten einen echten Mehrwert (teilweise rechtliche und steuerliche Äquivalenz zur händischen Unterschrift).

Exkurs 2: Zertifikatstypen

- Es gibt also:
  - **Einfache Zertifikate**
    z.B. selbstausgestellt: Ermöglichen Verschlüsselung, aber keinerlei Identitätsnachweis.
  - **Fortgeschrittene Zertifikate**
    z.B. von einer zertifizierten Institution für ihre Mitarbeiter ausgestellt: Ermöglichen Verschlüsselung und eine Plausibilisierung der Identität
  - **Qualifizierte Zertifikate**
    z.B. von einem Trustcenter ausgestellt: Institution hat Identität geprüft, Sperrung möglich (unterliegen dem SigG)
  - **Akkreditierte Zertifikate**
    von einem zertifizierten Trustcenter ausgestellt: Rechtlich sind die Angaben denen des Meldeamtes gleichgestellt

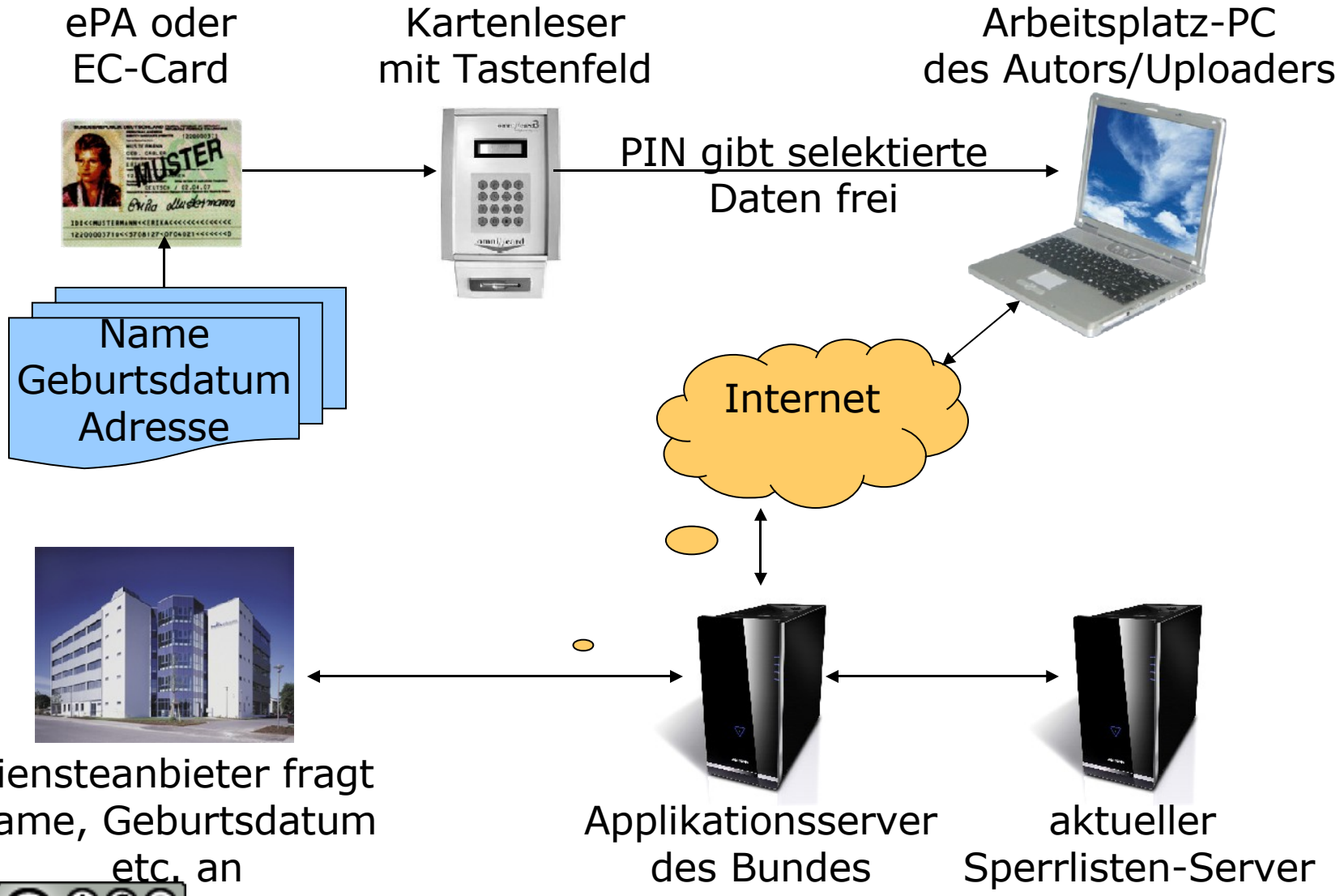
Exkurs 2: Zertifikat-Infrastruktur

- Aktuell betreiben nur wenige ZDAs ein akkreditiertes CA (Certificate Autority = umgangssprachlich: Trustcenter)
  - TeleSec (T-Systems) seit 12/1998
  - Bundesnotarkammer seit 12/2000
  - DATEV seit 3/2001
  - D-Trust seit 3/2002
  - Deutsche Post seit 9/2004
  - TC TrustCenter seit 5/2006
- S-Trust hat den Betrieb nur angezeigt, sich bisher nicht akkreditiert.
- Geplant (für den ePA = elektronischen Personalausweis – geplant ab 2008): BSI = Bundesamt für Sicherheit in der Informationstechnik [c't 12/2007 pp.72]

Exkurs 2: Zertifikate – welche?

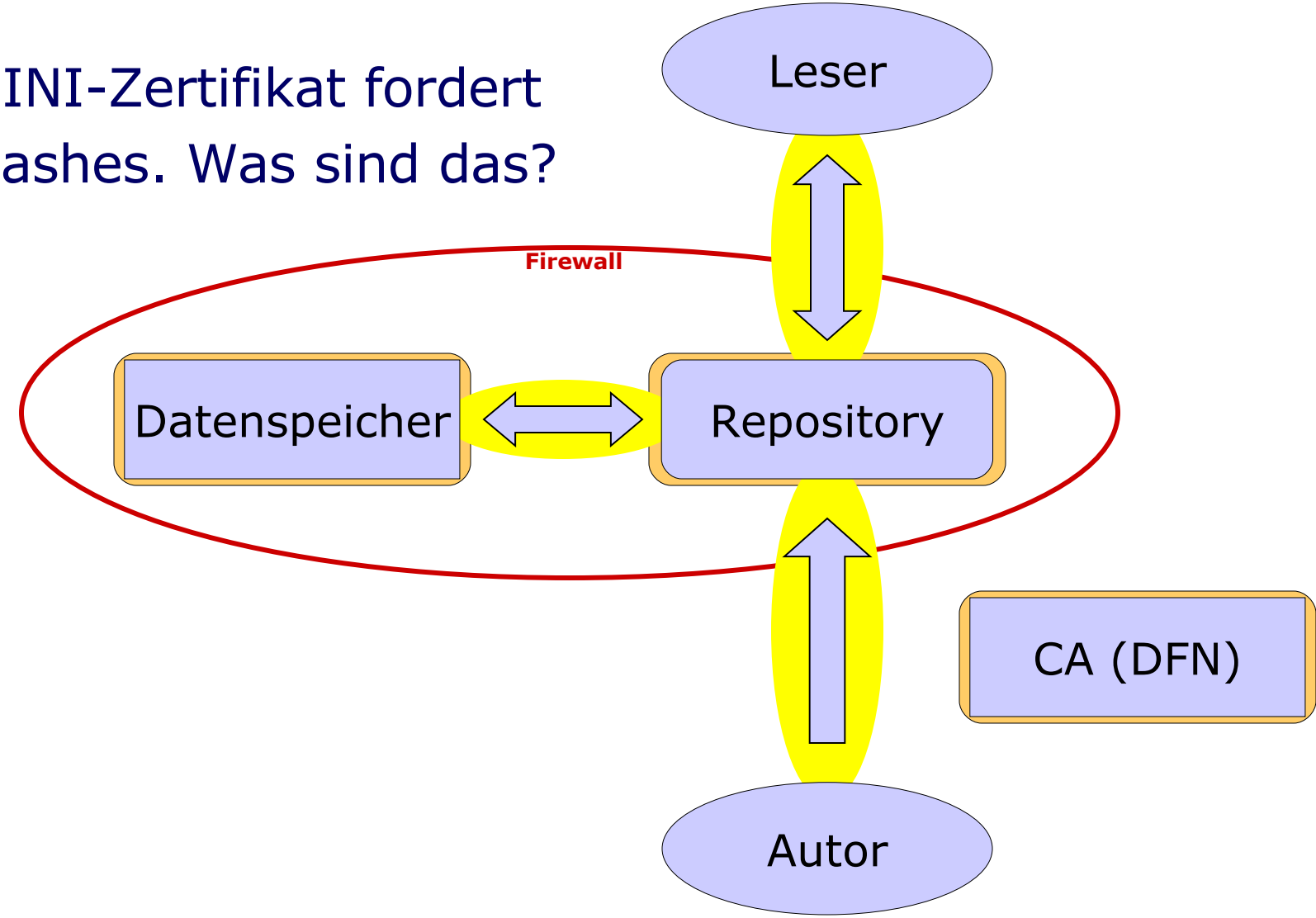
- Repositories sollten Identität garantieren:
  => Qualitifizierte Zertifikate
- Autoren sollten beim Upload ihre Identität nachweisen, um Missbrauch zu vermeiden:
  => Akkreditierte Zertifikate z.B. aus EC-Cards oder ePA (Infrastruktur steht erst Ende 2008 zur Verfügung)
  => „Work-arround" bis 2010: Autor unterschreibt ein Papier (bspw. die Rechteübertragung, siehe Workshop am 6.6.), Upload wird erst mit dieser Unterschrift freigeschaltet (Nachteil: Medienbruch).

Exkurs 2: Zertifikate: Identifikation mittels EC-Card oder ePA



Wir erinnern uns: Repository-Struktur

- DINI-Zertifikat fordert Hashes. Was sind das?



Exkurs 3:

Hashfunktionen
SHA-1 (Secure Hash Algorithm)
published by NIST (National Institute of Standards and Technology)

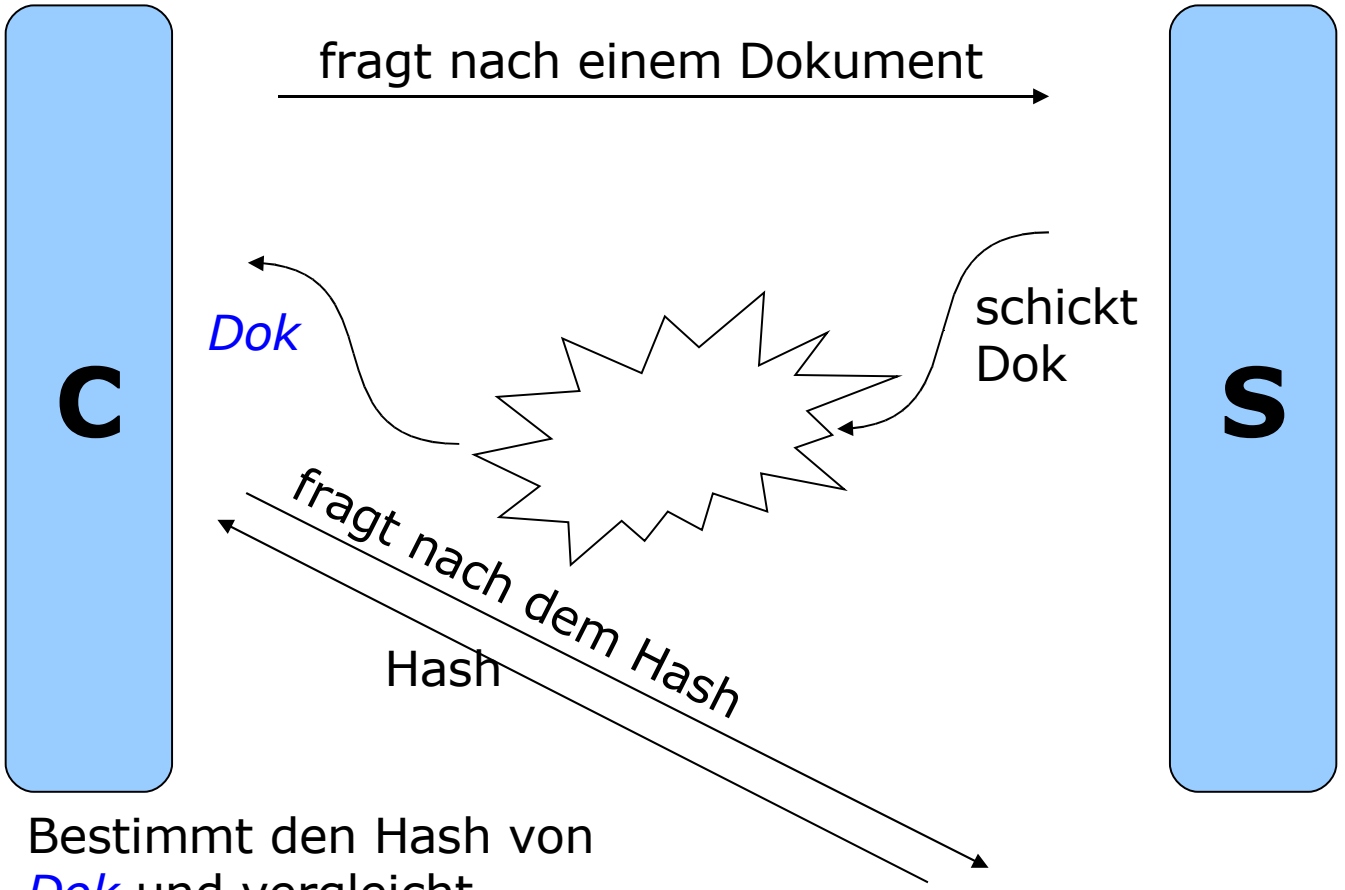
Übersicht

- Zweck von Hashfunktionen
- Mathematische Definition
- Wie wird ein SHA-1 Wert berechnet?
- Beispiele
- Eigenschaften und aktuelle Forschungsergebnisse

Wozu sind Hashfunktionen gut?

- Generierung eines „digitalen Fingerabdrucks“ oder einer Signatur
- Zur Verifizierung der Integrität von Dokumenten
- In Verbindung mit Zertifikaten zur Authentifizierung der Kommunikation bzw. des Kommunikationspartners

Beispiel



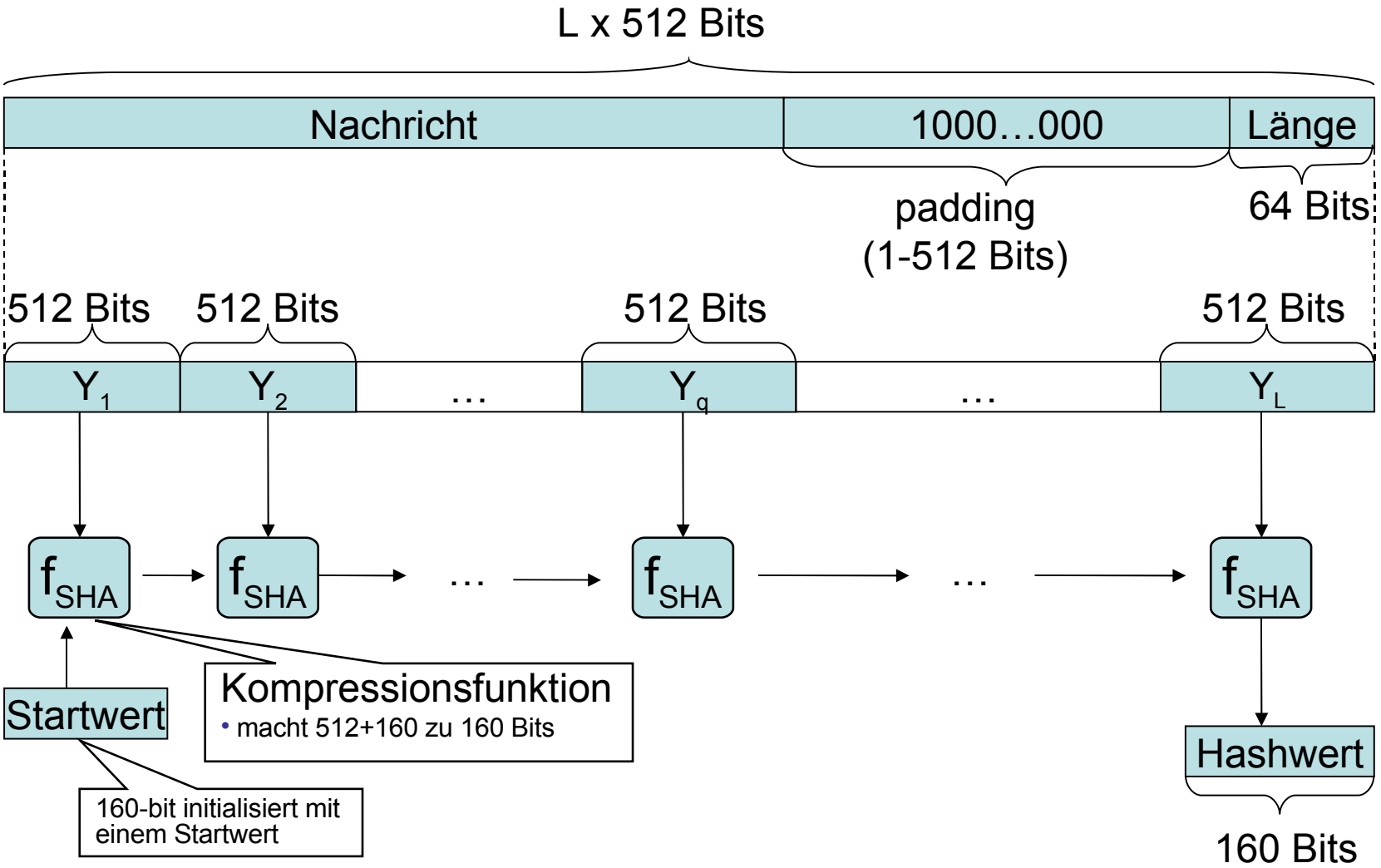
Was leisten Hashes in der Praxis?

- Zeigen Veränderungen der Daten an
  - durch bewusste Manipulation
  - durch technische Fehler
- Erlauben automatisches Prüfen des Datenbestandes auf Veränderungen (Monitoring)
- Ermöglichen Beweis der Unversehrtheit der Daten
  - Repository Betreiber darf Objekte *inhaltlich* nicht verändern.

Was ist eine Hashfunktion?

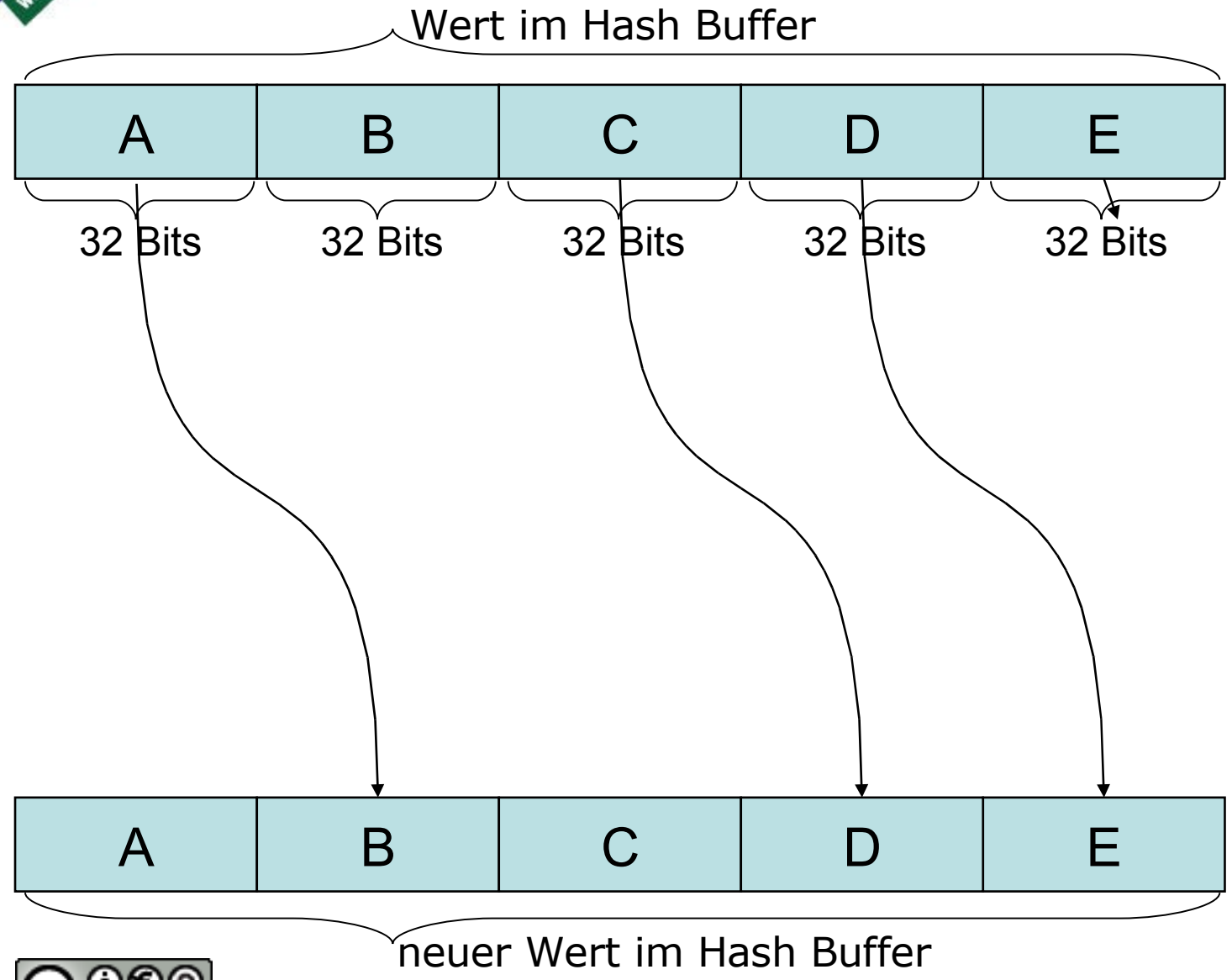
- Eine kryptographische Hashfunktion ist eine Abbildung $H:\{0,1\}^* \rightarrow \{0,1\}^n$ mit folgenden Eigenschaften:
  - H kann effizient berechnet werden
  - (preimage resistance) Die einzige Möglichkeit, zu einem gegebenen Hashwert y ein Dokument x mit H(x)=y zu finden, ist das Probieren von beliebigen Dokumenten (Brute force).
  - (collision resistance) Es ist effizient nicht möglich verschiedene Dokumente mit gleichem Hashwert zu finden. (Birthday attack)

Prinzipielle Idee von SHA-1



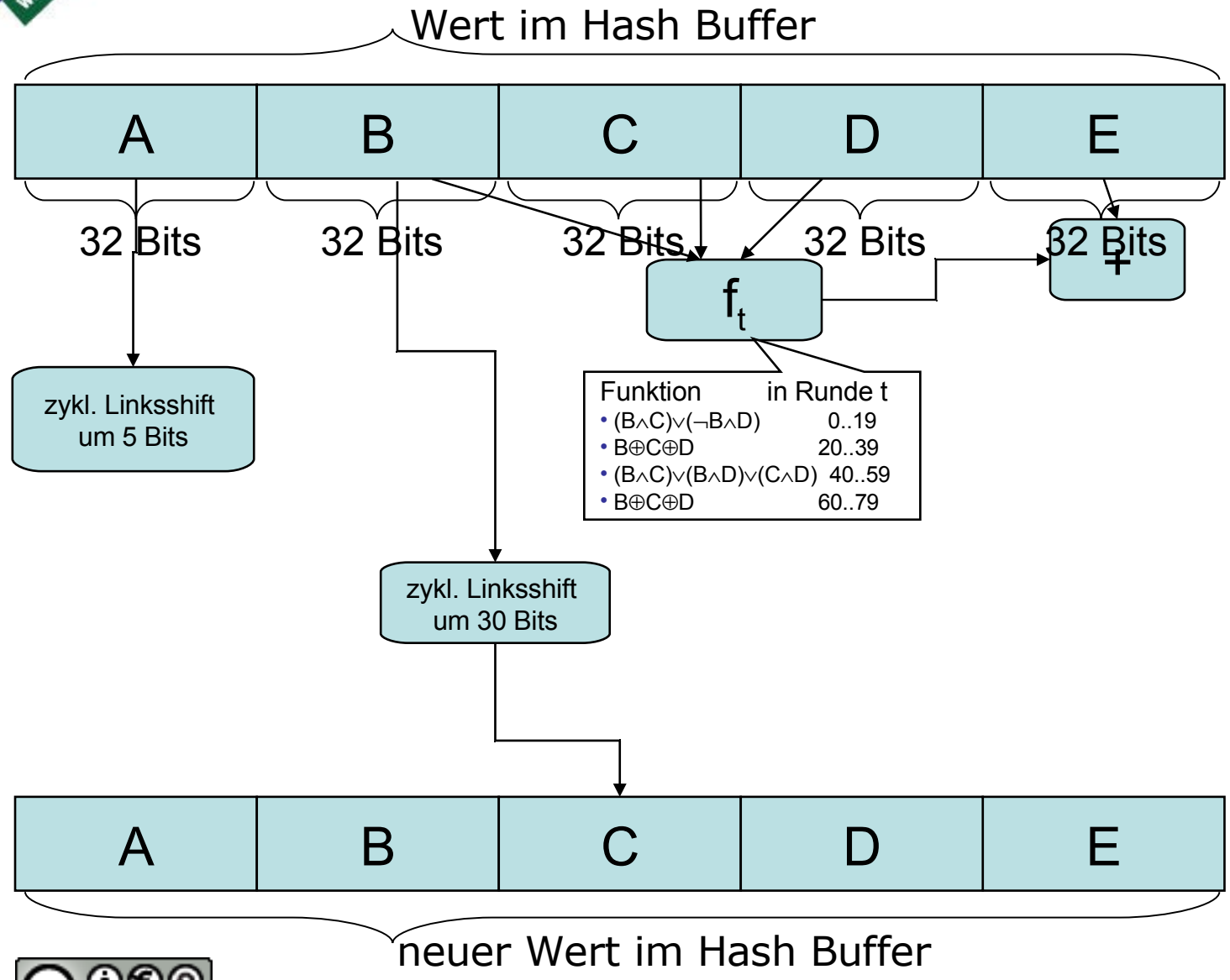
Ein Schritt f_{SHA}

(4 Runden á 20 Schritte)



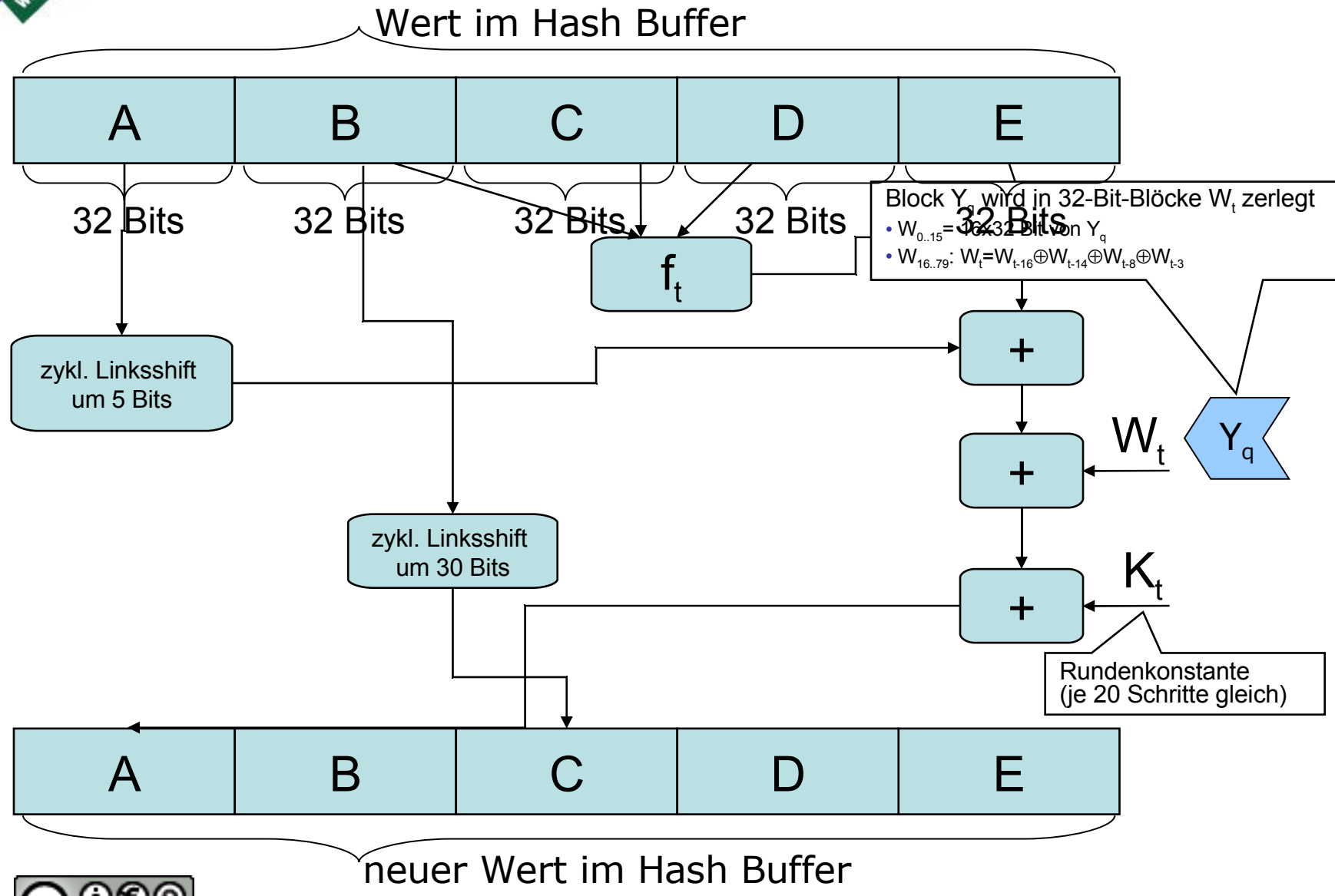
Ein Schritt f_{SHA}

(4 Runden á 20 Schritte)



Ein Schritt f_{SHA}

(4 Runden á 20 Schritte)



Beispiele

The quick brown fox jumps over the lazy dog
2fd4e1c67a2d28fced849ee1bb76e7391b93eb12

The quick brown fox jumps over the lazy **c**og
de9f2c7fd25e1b3afad3e85a0bd17d9b100db4b3

(Die 160 Bits des Hashwertes werden hier als Hexadezimal-Zahl dargestellt.)

Eigenschaften

- Jedes Outputbit hängt von jedem Inputbit ab
  - wichtig für Kollisionsresistenz
- Preimage in 2^{160} Versuchen (brute force)
- Collision in 2^{80} Versuchen (birthday attack)
- Mittlerweile ist die Bestimmung von Collisionen mit 2^{69} Versuchen möglich (2005, China)
- Generierung von Hashzwillingen, deren Inhalt zu 25% wählbar ist (der Rest ist dann Buchstabensalat).

Fazit

- Auch wenn SHA-1 die Kriterien einer echten Hashfunktion nicht erfüllt, so ist sie doch für praktische Anwendungen wie TLS oder Schutz vor Datenmutation bisher ausreichend sicher.

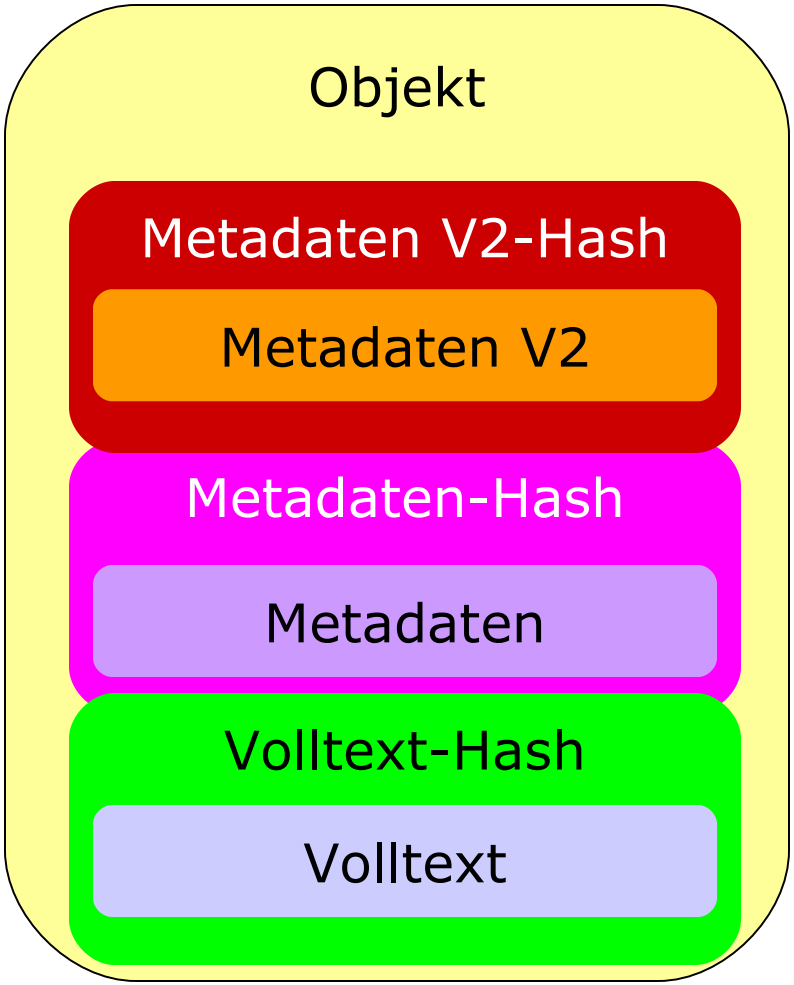
Was leisten Hashes in der Praxis?

- Zeigen Veränderungen der Daten an
  - durch bewusste Manipulation
  - durch technische Fehler
- Erlauben automatisches Prüfen des Datenbestandes auf Veränderungen (Monitoring)
- Ermöglichen Beweis der Unversehrtheit der Daten
  - Repository Betreiber darf Objekte *inhaltlich* nicht verändern.

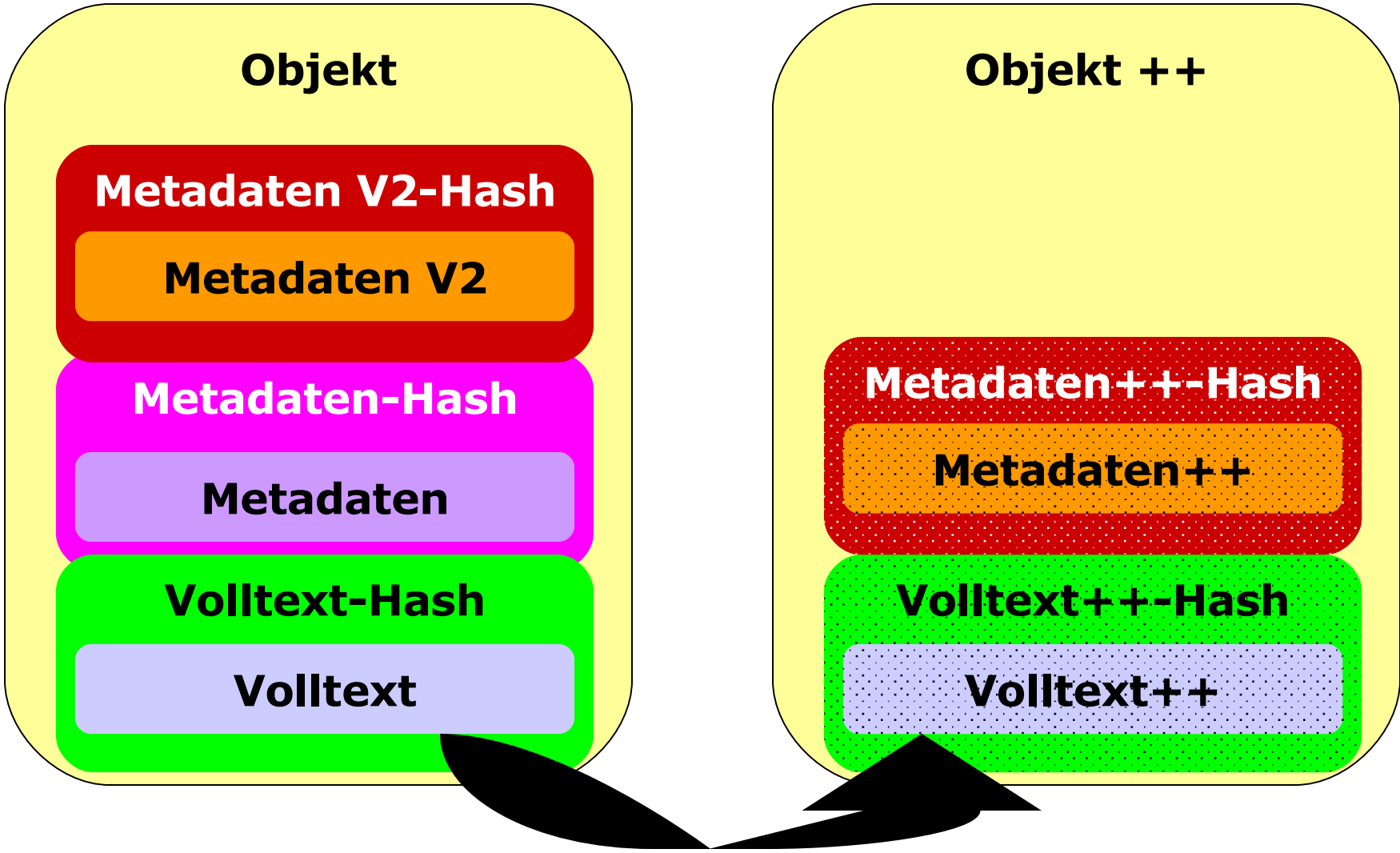
Hashes in der Praxis

- Repository Software erzeugt Hashes
  - beim Einstellen der Objekte
  - bei Erzeugung neuer Objekte (z.B. Erstellen einer neuen Version im Rahmen der Migration auf andere Dateiformate)
- Hashes getrennt für Metadaten und für Volltext
  - Metadaten ändern sich z.B. wenn Referenzen hinzugefügt werden können (ehem. gedruckte Artikel sind online verfügbar, also verlinkbar)
- Hashes liegen als Teil der Objekte im Datenspeicher

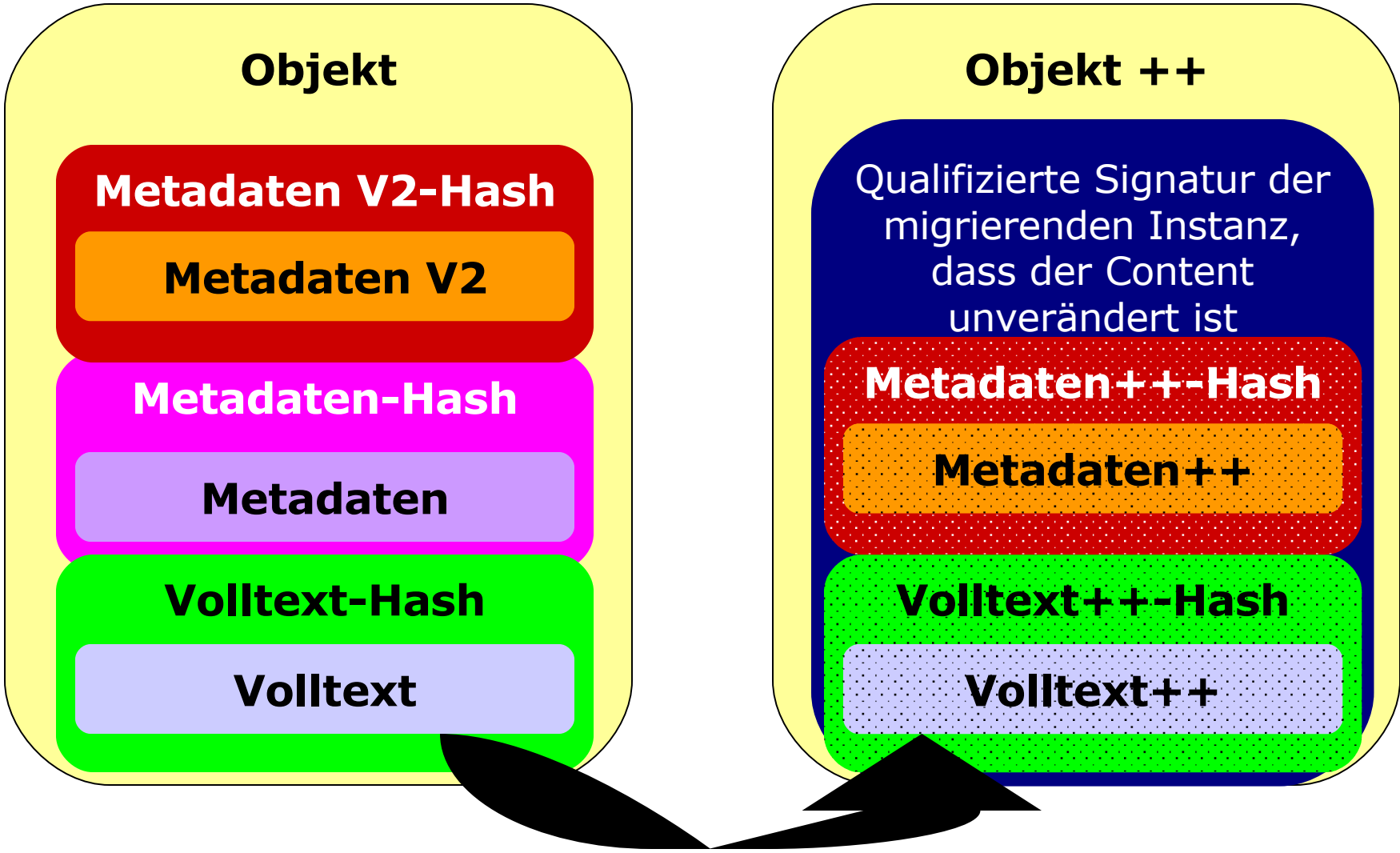
Objekte im Datenspeicher



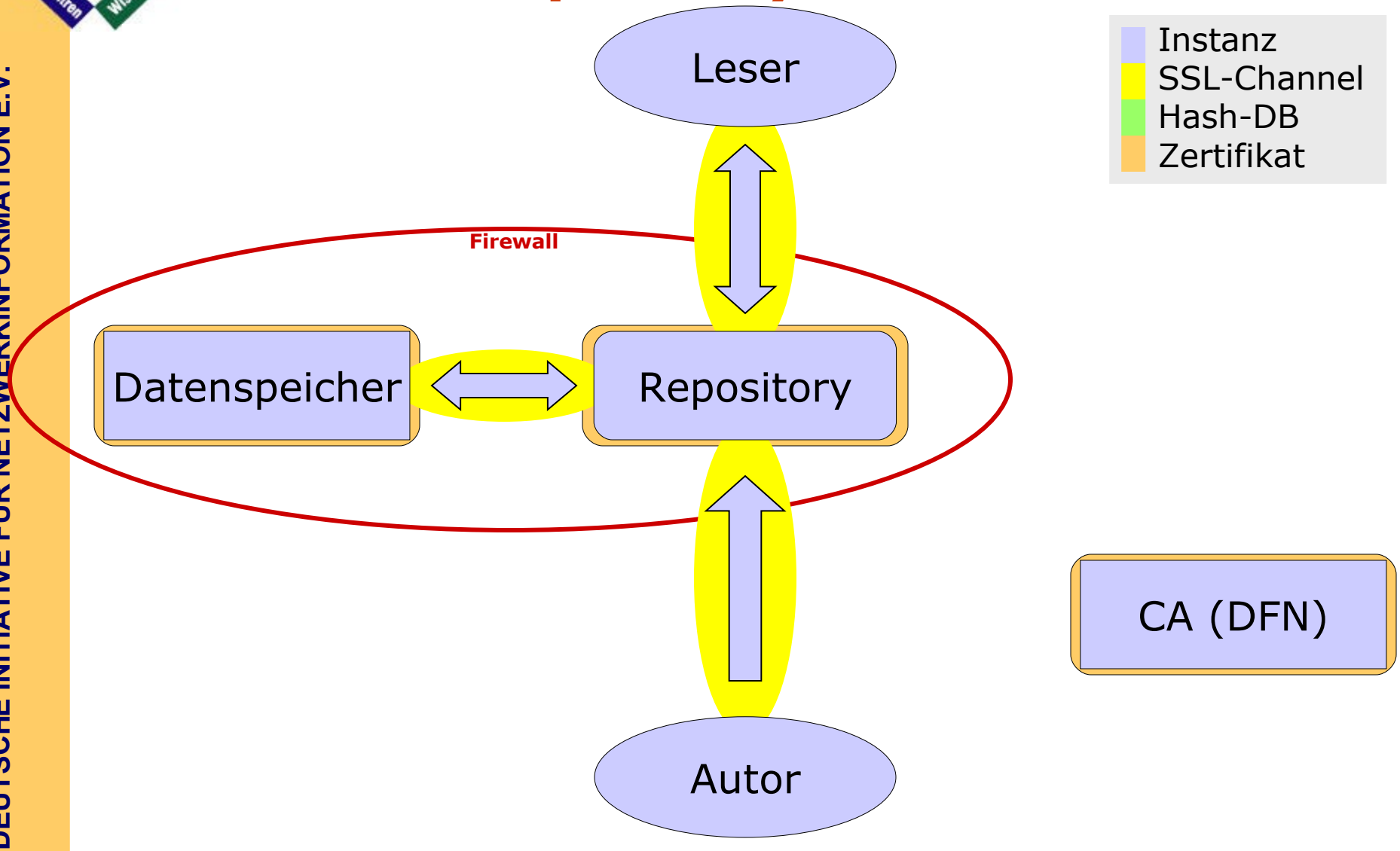
Migration der Objekte



Migration der Objekte



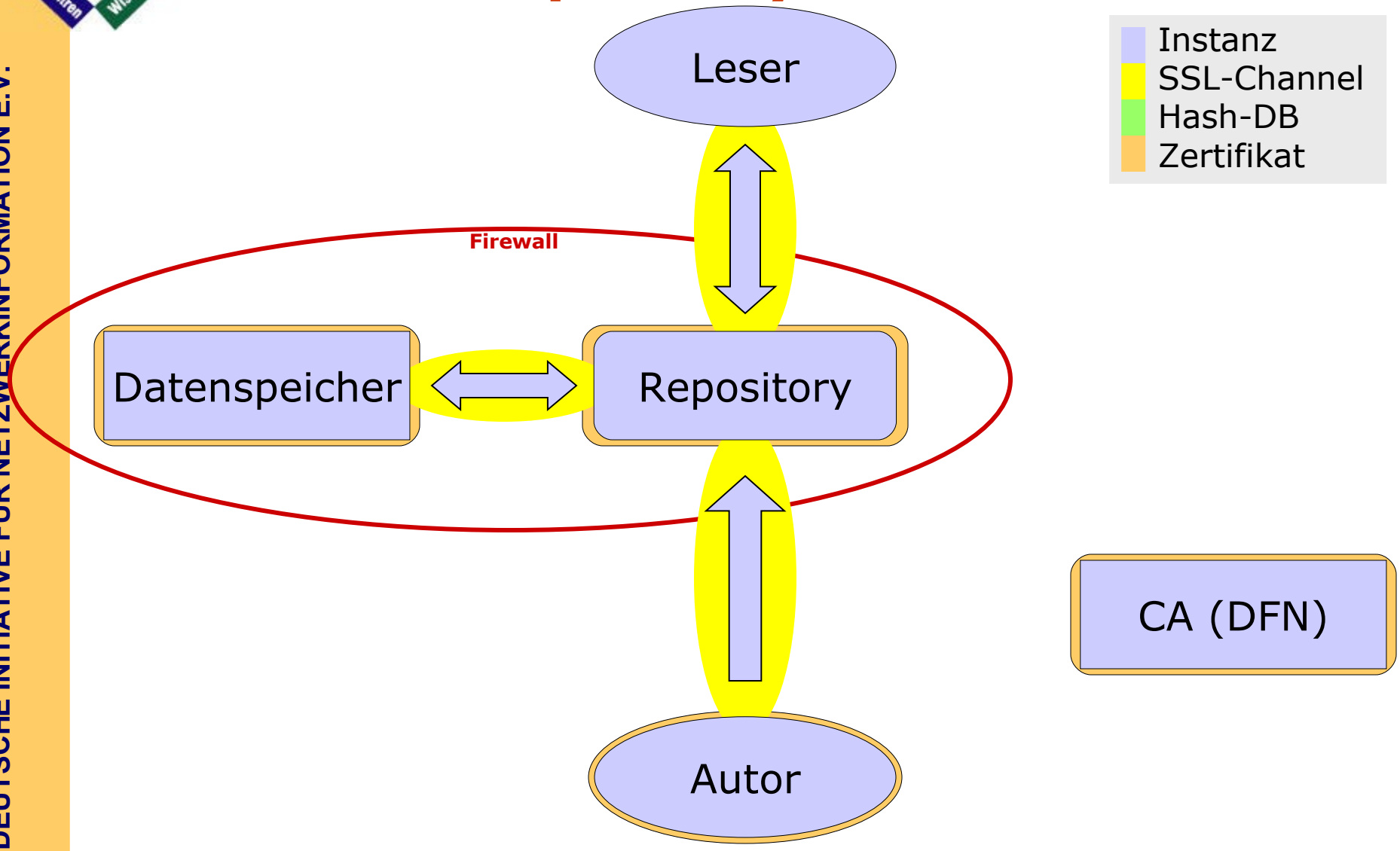
Integration der Anforderungen in die Repository Infrastruktur



Integration der Anforderungen in die Repository Infrastruktur

- Autor sollte sich authentifizieren durch qualifizierte oder akkreditierte Signatur
- Alternativ: Schriftliche Erklärung des Autors und temporäre Quarantäne der hochgeladenen Objekte (Medienbruch)

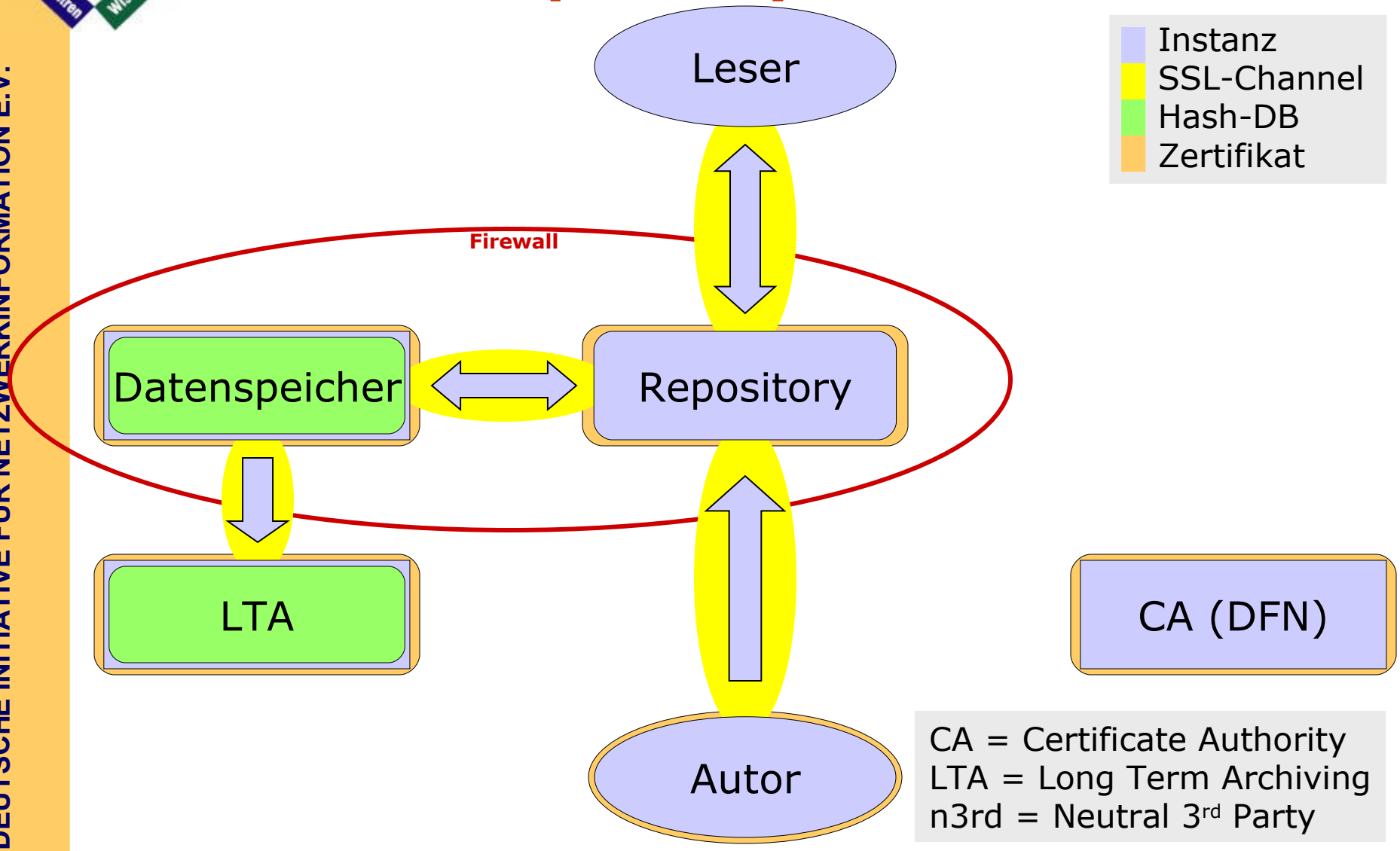
Integration der Anforderungen in die Repository Infrastruktur



Integration der Anforderungen in die Repository Infrastruktur

- Garantie der Integrität der Objekte im Datenspeicher mittels Hashes
- Export von Objekten aus dem Datenspeicher an eine externe Instanz zur Langzeitarchivierung (LTA)

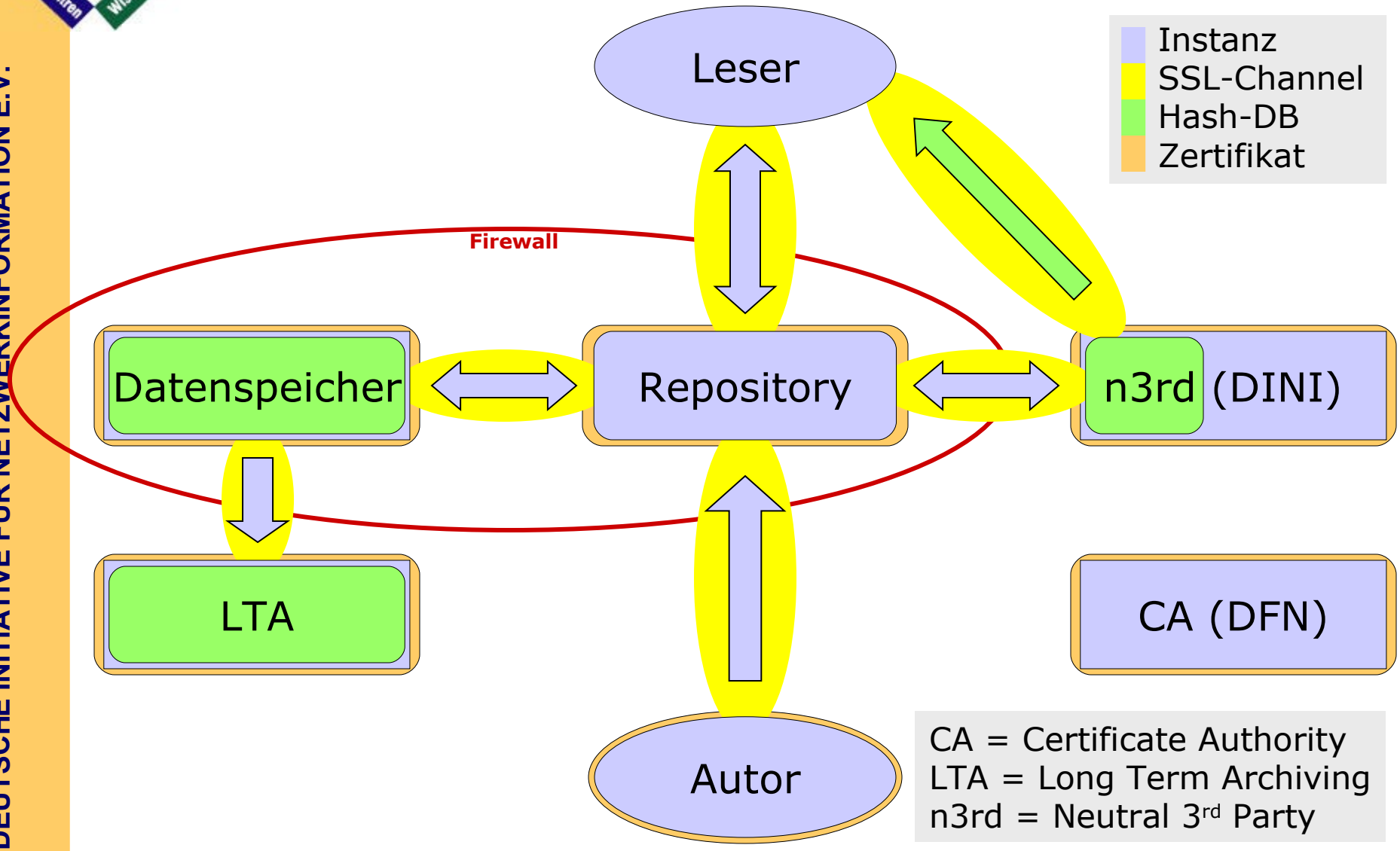
Integration der Anforderungen in die Repository Infrastruktur



Integration der Anforderungen in die Repository Infrastruktur

- Leser soll prüfen können, ob die Hashes im Repository seit dem Upload der Objekte verändert wurden (Prüfung, ob die im DINI-Zertifikat geforderte Strategie: „Jede Veränderung eines Objektes resultiert in einem neuen Objekt“ dauerhaft eingehalten wurde).
- Möglichkeit eines automatisierten Audithing der Datenspeicher und Repositories.
- Schaffung einer unabhängigen Instanz (z.B. DINI), die Kopien aller Identifier und Hashes anbietet.

Repository Infrastruktur



Empfehlungen

- Alle Kommunikationskanäle sollen mit SSL/TLS verschlüsselt und alle Instanzen durch eine CA qualifiziert zertifiziert sein.
- Pro Dokument sollte der hochladende (Autor) sich schriftlich authentifizieren (im Rahmen eines Lizenzvertrages), Umstellung auf ein online-Verfahren in 2010 einplanen.
- Getrennte Erzeugung von Hash-Werten der Volltexte und Metadaten.
- LTA-Instanz soll bei Migration, inhaltliche Identität durch qualifizierte Signatur garantieren.
- Export von Hash-Werten an eine unabhängige, vertrauenswürdige Institution.