Karlsruher Institut für Technologie

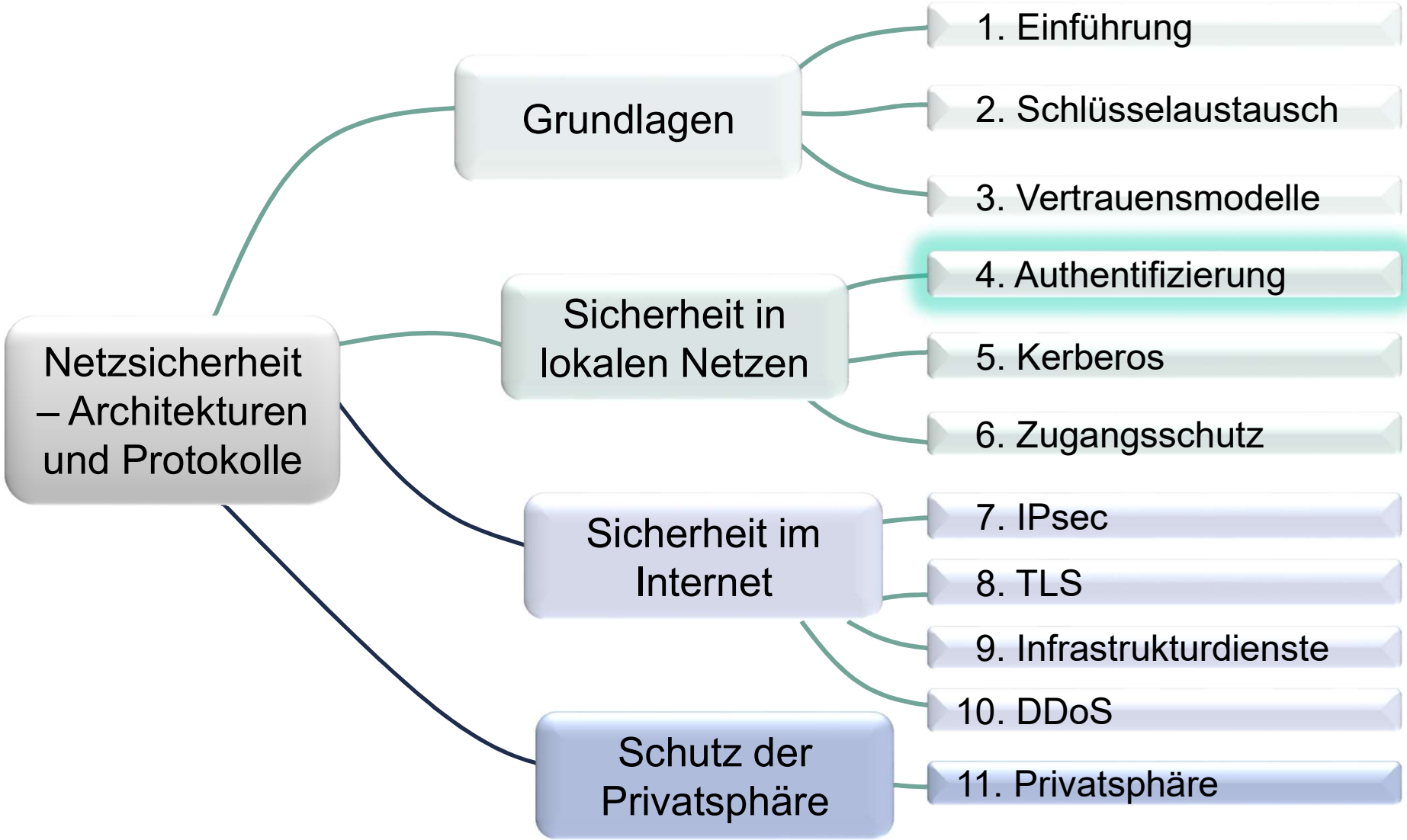
Vorlesung Netzsicherheit
Kapitel 4 – Authentifizierung

PD Dr. Ingmar Baumgart, PD Dr. Roland Bless, Matthias Flittner, Prof. Dr. Martina Zitterbart
baumgart@fzi.de, [bless, flittner, zitterbart]@kit.edu

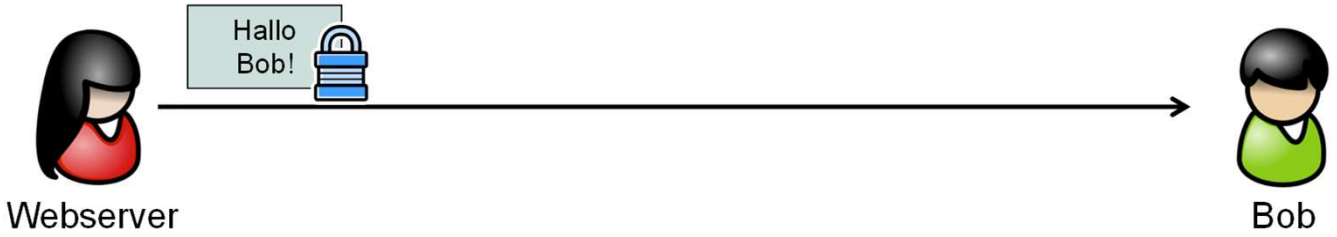
Institut für Telematik, Prof. Zitterbart

© Peter Baumung

Inhalte der Vorlesung



Bisher: Authentifizierung mittels Zertifikaten



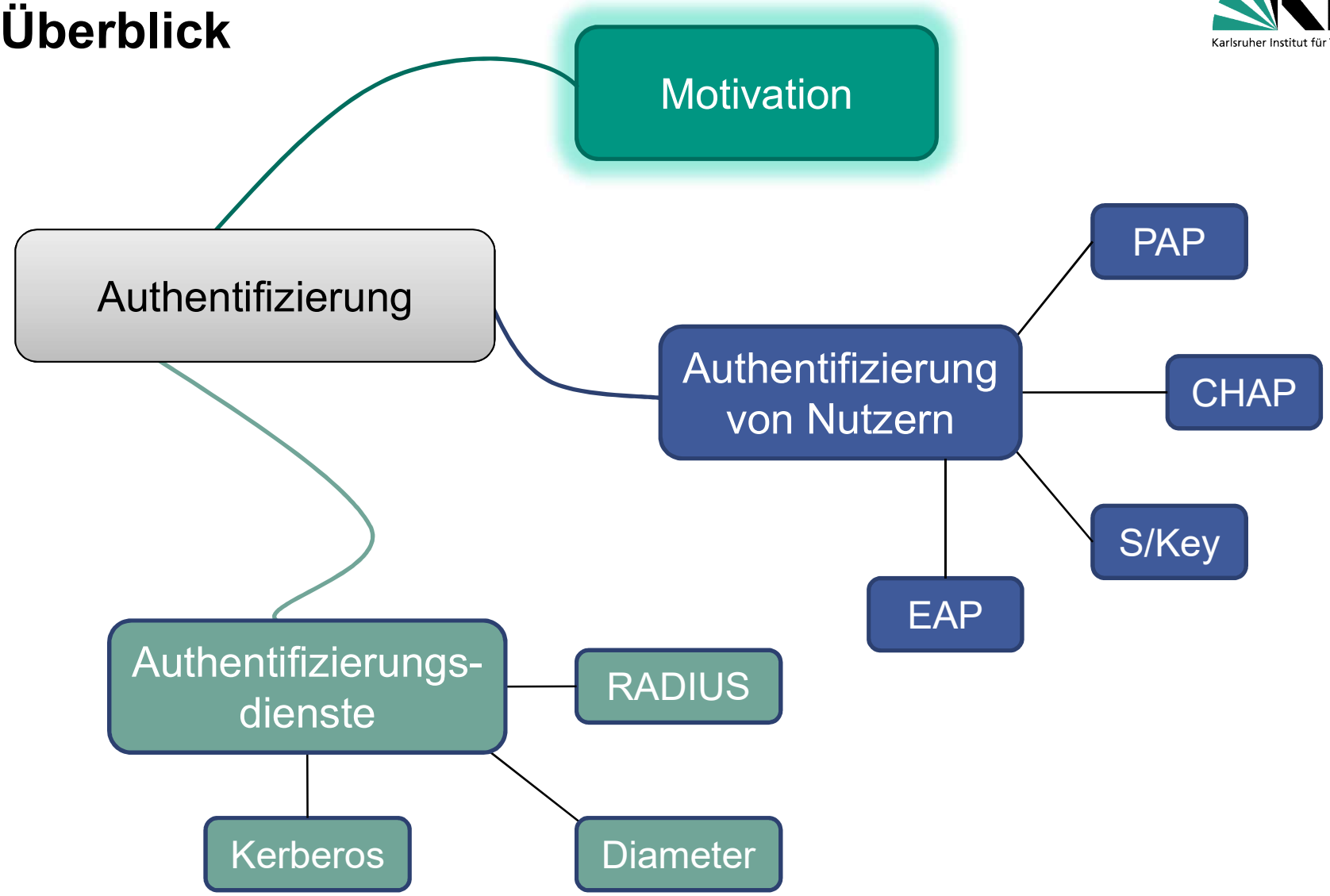
- Zum Beispiel: „*gesicherte*“ Kommunikation mit Webserver
  - Zunächst: Initialisierung einer „ungesicherten“ TCP-Verbindung
  - Authentische Aushandlung von Schlüsselmaterial (TLS)
  - Authentifizierung des Webservers mittels Zertifikat (TLS)
  - Validierung des Zertifikates bei Alice (TLS)
  - ➔ „*Gesicherter*“ Kanal zwischen Alice und Webserver

- Überprüfung der „*Echtheit*“ auf CA ausgelagert!
  - CA bescheinigt, dass Webserver zur Betreiberdomäne gehört
  - Ermöglicht Kommunikation mit zunächst unbekannten Webservern
  - So muss nicht jeder Nutzer separat die „Echtheit“ überprüfen

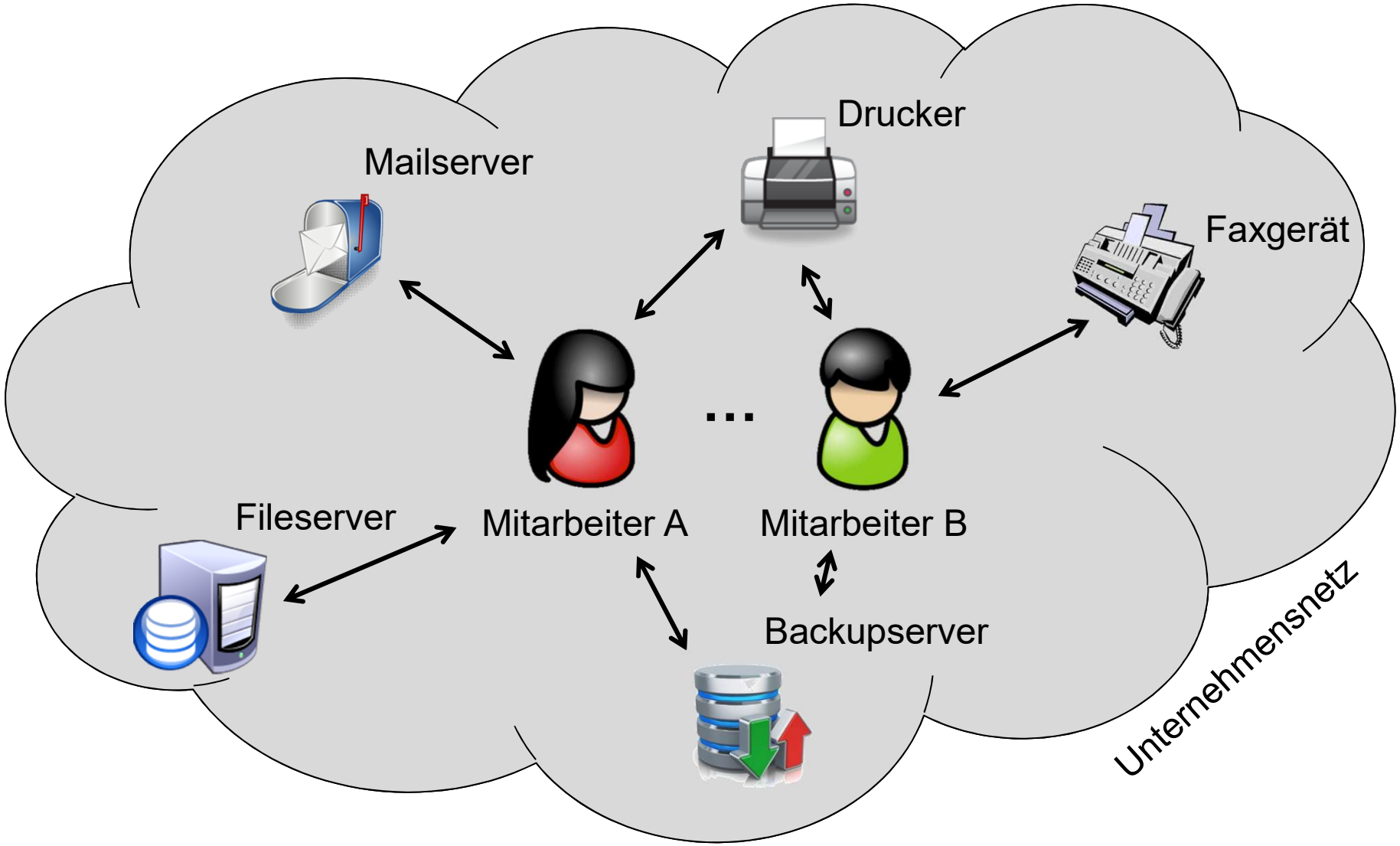
Jetzt: Passwort-basierte Authentifizierung

- Authentifizierung gegenüber Dienstbetreibern
  - z. B. bei Internetzugang, Onlinebanking, Mail, Druckdienst
- Vorgelagerte Registrierung / Vertragsaushandlung
  - Meist Vereinbarung eines gemeinsamen Geheimnisses (Passwort)
  - Muss separat abgesichert werden
    - Meist über einen zweiten Faktor (SMS, Brief, Postident, Treffen)
- Ziele des Dienstbetreibers
  - Nutzer eindeutig identifizieren
  - Illegitime Nutzer so früh wie möglich ausschließen
  - Nutzern nur die vereinbarten Rechte gewähren
  - Rechnungslegung

Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
PAP
CHAP
S/Key
EAP
Authentifizierungs-dienste
RADIUS
Kerberos
Diameter

Beispiel: Im Unternehmen

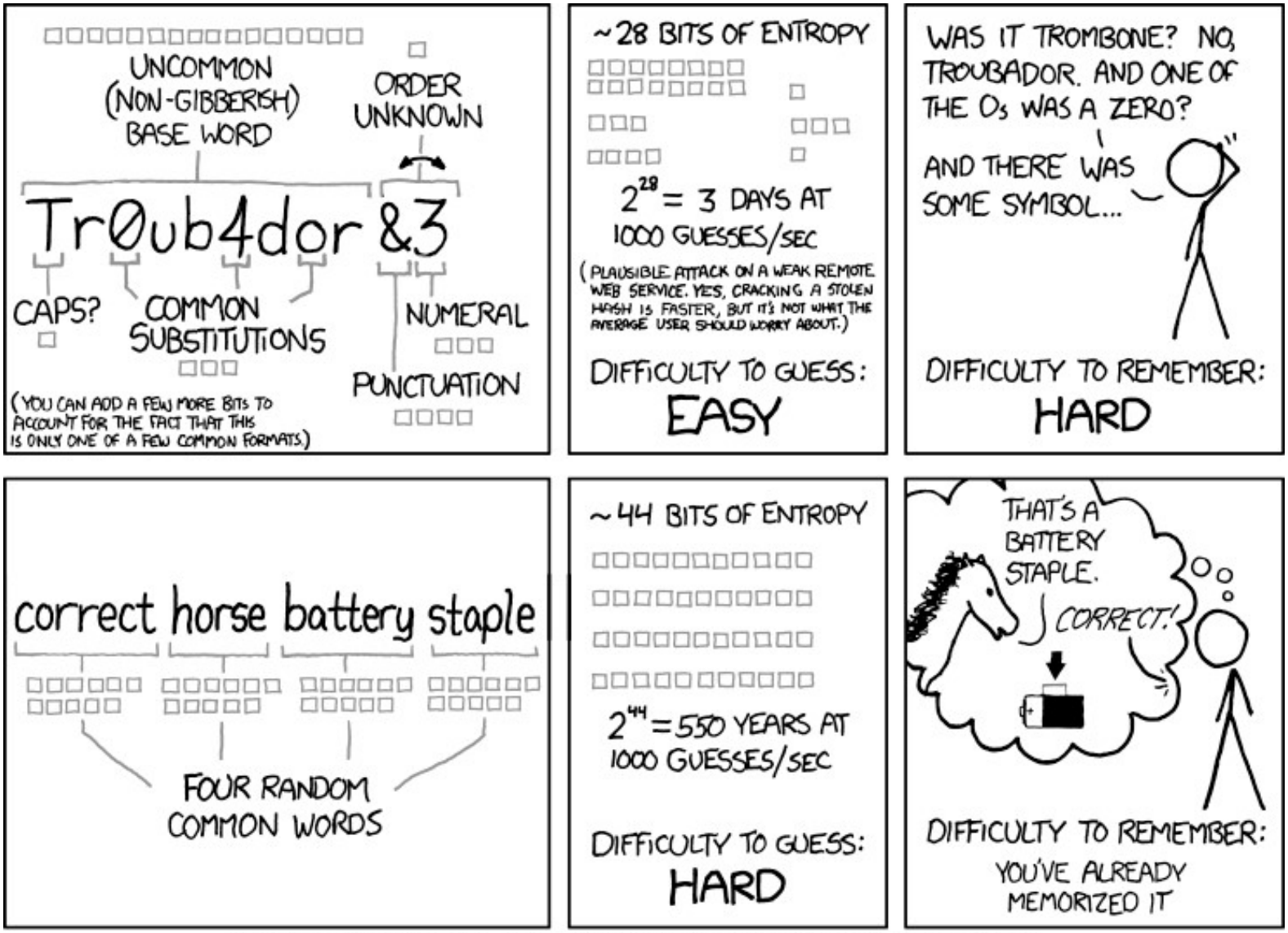


Aufgaben des Dienstbetreibers: AAA

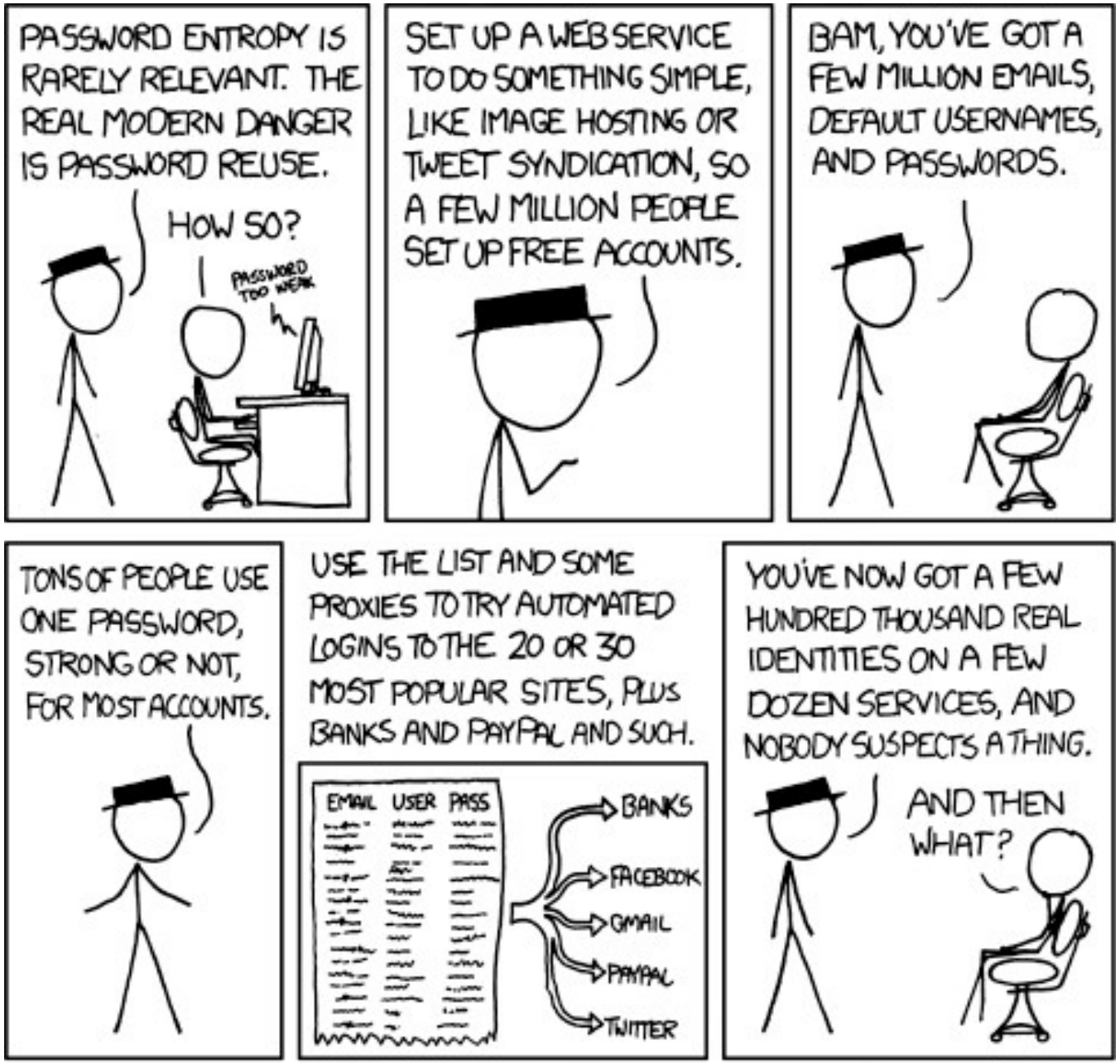
- Authentifizierung (engl.: *Authentication*)
  - Überprüfung, ob Kommunikationspartner tatsächlich derjenige ist, der er vorgibt zu sein
  - Besitz, Wissen, Biometrisches Merkmal
- Autorisierung (engl.: *Authorization*)
  - Überprüfen von vereinbarten Zugriffsrechten (z.B. Lesen bzw. Schreiben auf Fileserver)
  - Direkt im Anschluss zur Authentifizierung
- Abrechnung (engl.: *Accounting*)
  - Erfassung des Nutzungsverhaltens
  - Zum Beispiel nach Dauer, Anzahl, etc.


[RFC3539]

Wie wählt man ein sicheres Passwort und worauf muss man dabei achten?

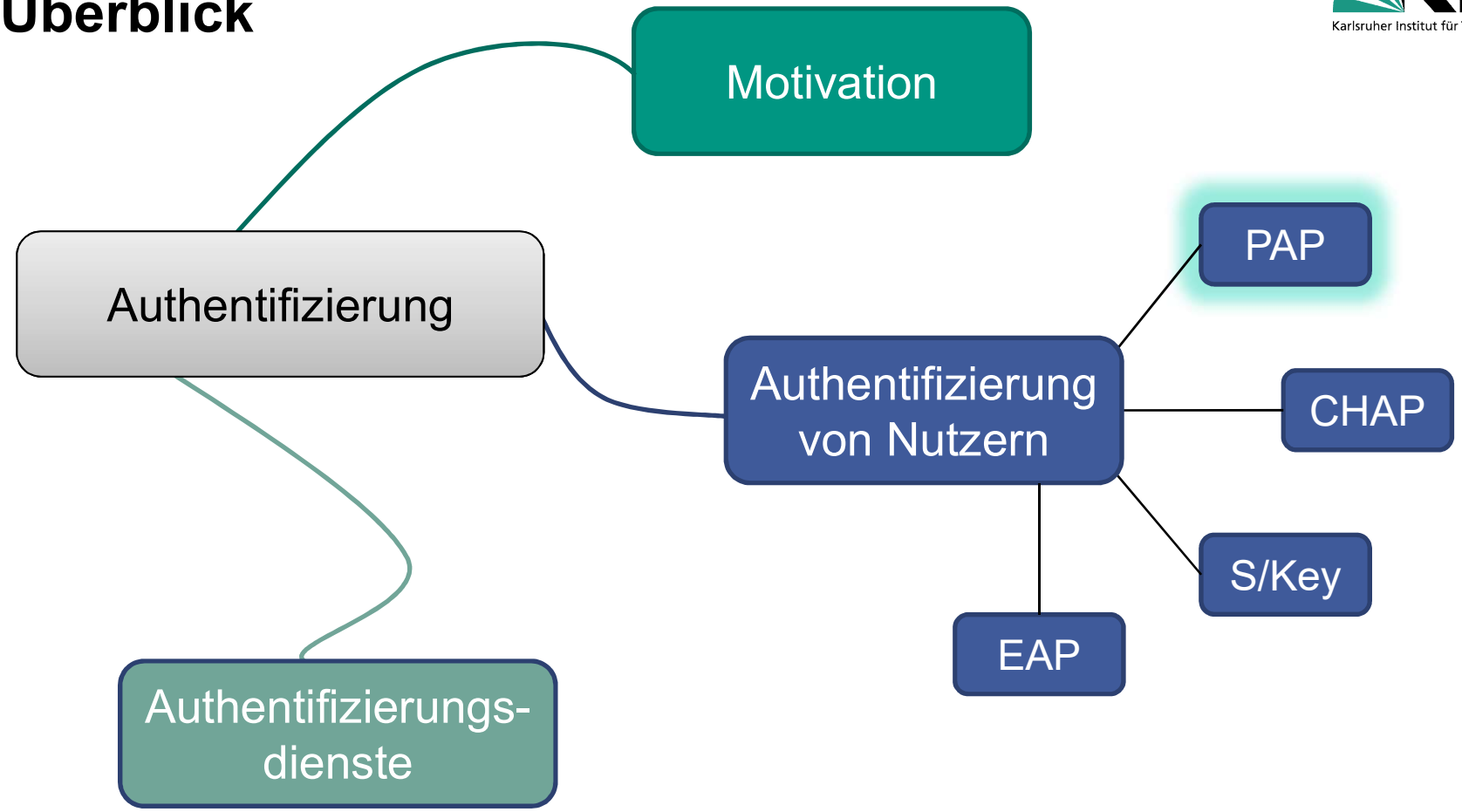

UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Quelle: xkcd


PASSWORD ENTROPY IS RARELY RELEVANT. THE REAL MODERN DANGER IS PASSWORD REUSE.
HOW SO?
PASSWORD TOO WEAK
SET UP A WEB SERVICE TO DO SOMETHING SIMPLE, LIKE IMAGE HOSTING OR TWEET SYNDICATION, SO A FEW MILLION PEOPLE SET UP FREE ACCOUNTS.
BAM, YOU'VE GOT A FEW MILLION EMAILS, DEFAULT USERNAMES, AND PASSWORDS.
TONS OF PEOPLE USE ONE PASSWORD, STRONG OR NOT, FOR MOST ACCOUNTS.
USE THE LIST AND SOME PROXIES TO TRY AUTOMATED LOGINS TO THE 20 OR 30 MOST POPULAR SITES, PLUS BANKS AND PAYPAL AND SUCH.
EMAIL
USER
PASS
BANKS
FACEBOOK
GMAIL
PAYPAL
TWITTER
YOU'VE NOW GOT A FEW HUNDRED THOUSAND REAL IDENTITIES ON A FEW DOZEN SERVICES, AND NOBODY SUSPECTS A THING.
AND THEN WHAT?

Quelle: xkcd

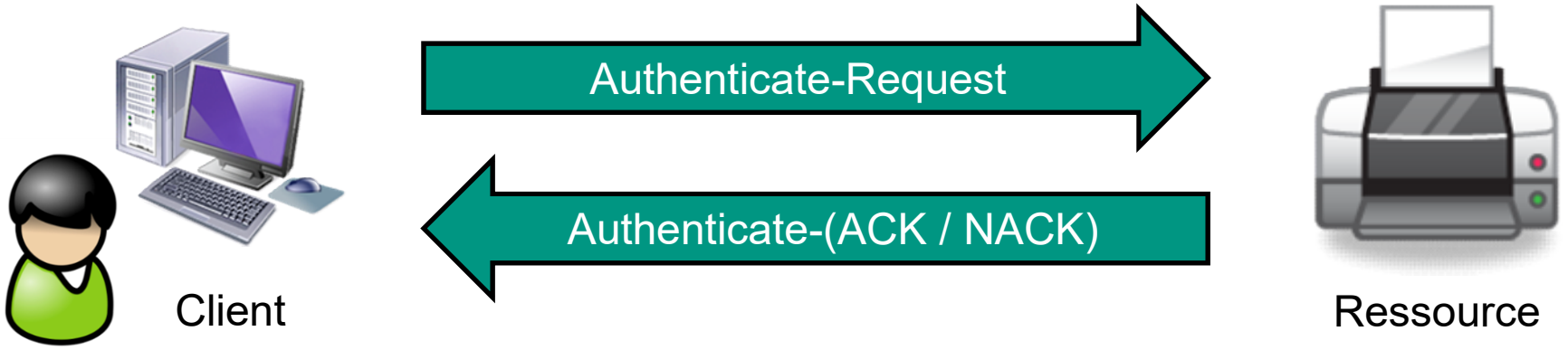
Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
PAP
CHAP
S/Key
EAP
Authentifizierungs-dienste

Password Authentication Protocol (PAP)

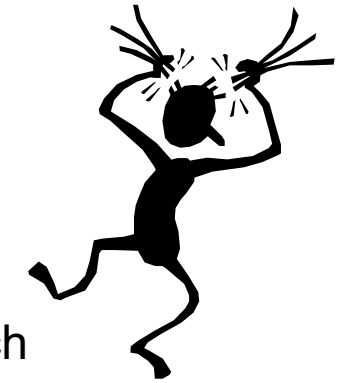
[RFC1334]

- Idee
  - Nutzer identifiziert sich mittels Nutzerkennung (ID) und dazugehörigem Passwort
- Ablauf
  - Nutzer gibt ID und Passwort auf Client ein
  - Client schickt ID / Passwort-Paar an Ressource
  - Ressource bestätigt oder lehnt ab



Password Authentication Protocol (PAP)

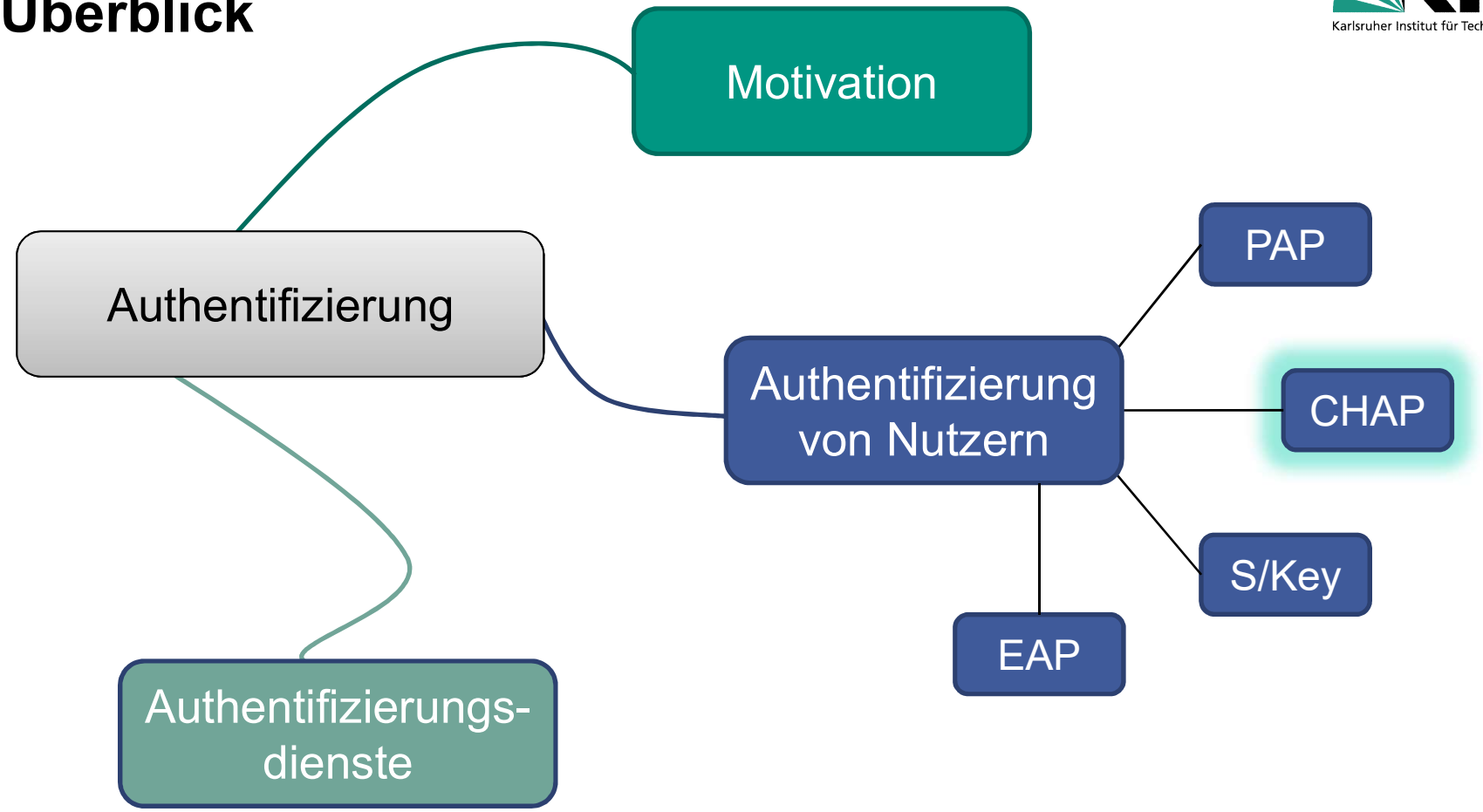
- Schwächen
  - Übertragung des Passworts im *Klartext!*
    - Man-in-the-Middle kann Passwörter abhören
    - Replay-Angriff möglich
  - Client ist Initiator
    - Kann Anfrage-Frequenz bestimmen; DoS-Angriffe möglich
  - Ressource hat Zugriff auf das Klartext-Passwort
    - Bei Einbruch auf Ressource werden alle Passwörter bekannt!
    - Ablage der Passwörter als Hash bietet gewissen Schutz - Ressource erhält aber dennoch bei Authentifizierungsanfrage Klartext-Passwort



➔ Natives PAP unsicher!

- Aber praktikabel über zuvor aufgebauten *gesicherten* Kanal
- Dazu muss sich jedoch zunächst Ressource gegenüber Client authentifizieren
  - Sonst wieder Möglichkeit für Man-in-the-Middle…

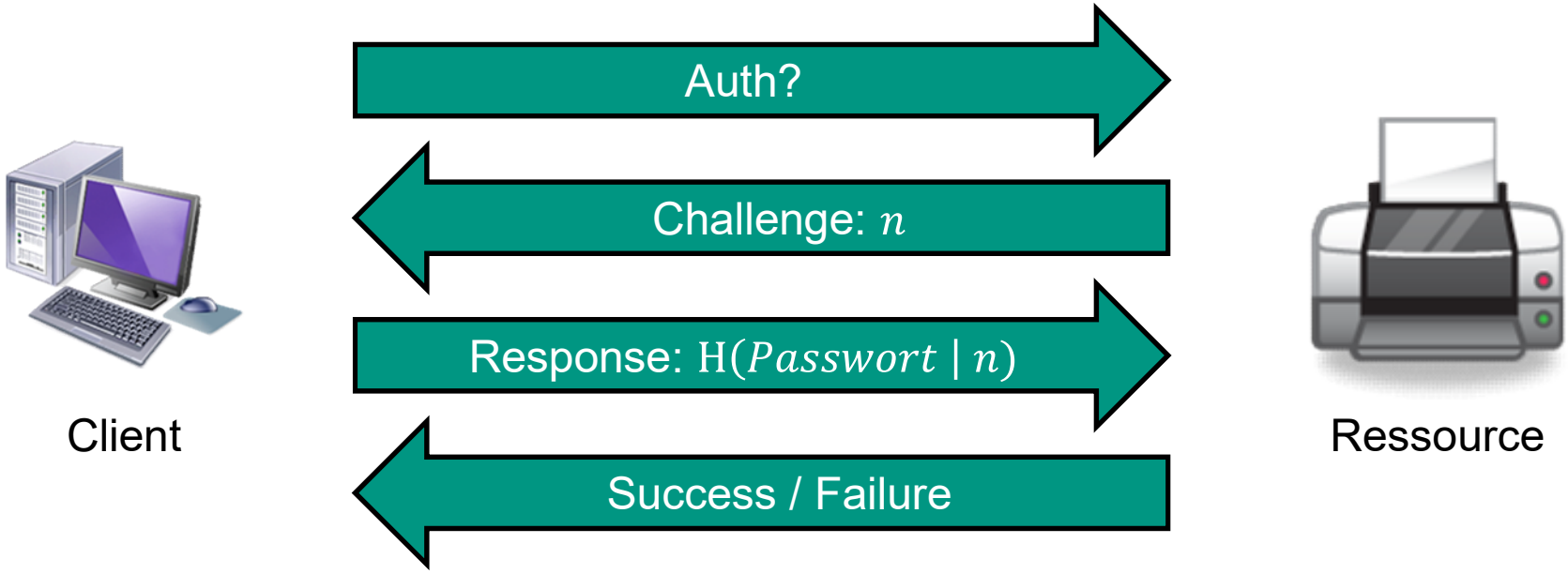
Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
PAP
CHAP
S/Key
EAP
Authentifizierungs-dienste

Challenge Handshake Authentication Protocol (CHAP)

- Grundlegendes Konzept
  - Keine direkte Übertragung von Passwörtern
    - Passwort-Hash gekoppelt mit Challenge

[RFC1994]



Bewertung CHAP

- Vorteile gegenüber PAP
  - Passwort nicht im Klartext übertragen
    - Replay-Angriff nicht möglich (bei guter Challenge)
  - Hash-Algorithmen frei wählbar: MD5, SHA-1, SHA-3, …
- Nachteil
  - Passwort auf der Ressource im Klartext gespeichert!
    - Bei Einbruch auf Ressource können diese ausgelesen werden

Implementierung PAP vs. CHAP

- Direkter Vergleich
  - Authentifizierung über ungesicherten Kanal

	Natives PAP	Natives CHAP
Empfang	Passwörter im Klartext	Gehashte Passwörter
Passiver Angreifer	Erfährt alle Passwörter	Lernt “nichts”
Nutzer-Datenbank	Hash	Klartext
Einbruch auf Ressource	Deckt “nur” neue Authentifizierungen auf	Deckt alle Passwörter auf
Sicherheit	Sicherheit der Hashfunktion	Sicherheit der Hashfunktion, Wahl der Nonces

Reality-Check: MS-CHAP v2

- Proprietäre Lösung von Microsoft
  - Jahr 1998 / 2000: Windows 95/98/NT4
  - Verwendung von Verschlüsselung anstatt Hashfunktion

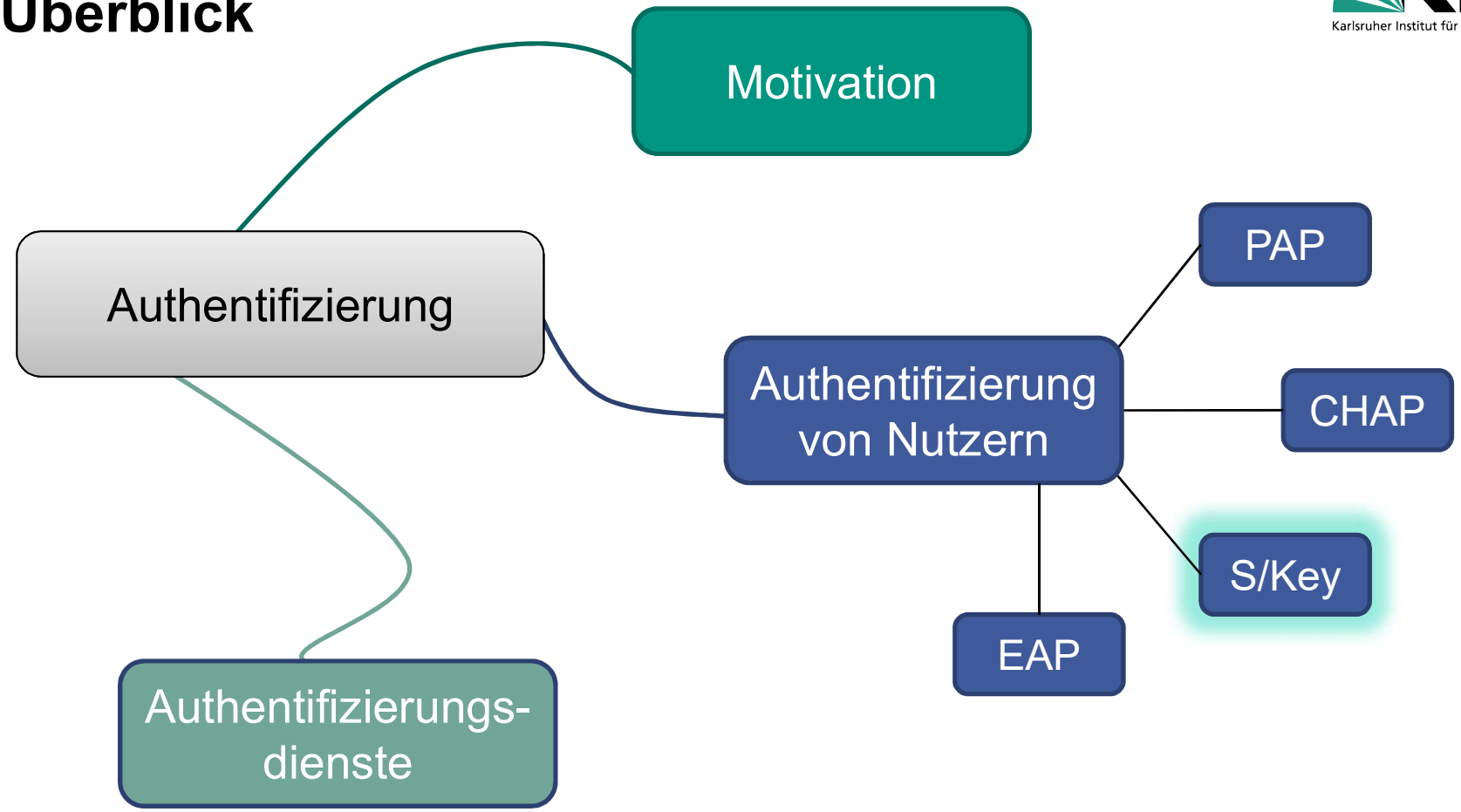
[RFC2759]

- Ablauf
  - Ressource sendet Challenge CH an Client
  - Client berechnet:
    - Passworthash $PH := H(Passwort)$
    - Response: $DES_{PH[0:6]}(CH)$ || $DES_{PH[7:13]}(CH)$ || $DES_{PH[14:20]}(CH)$

- Aber: DES ist gebrochen!
  - 2012: Online-Dienst (*cloudcracker*) zum Berechnen des PH
  - Mit heutigen GPUs in wenigen Tagen zu brechen

➔ Daher: MS-CHAP v2 nur noch über gesicherten Kanal!

Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
PAP
CHAP
S/Key
EAP
Authentifizierungs-dienste

S/Key - One-Time Password (OTP)

- Ziel
  - Verhindern von Replay-Angriffen
  - Keine Passwörter auf Ressource speichern
  - Verwendung von Einmalpasswörtern

S_0	$H(Password + Seed)$
S_1	$H(S_0)$
S_2	$H(S_1)$
...	...
S_n	$H(S_{n-1})$

[RFC1760] bzw. [RFC2289]

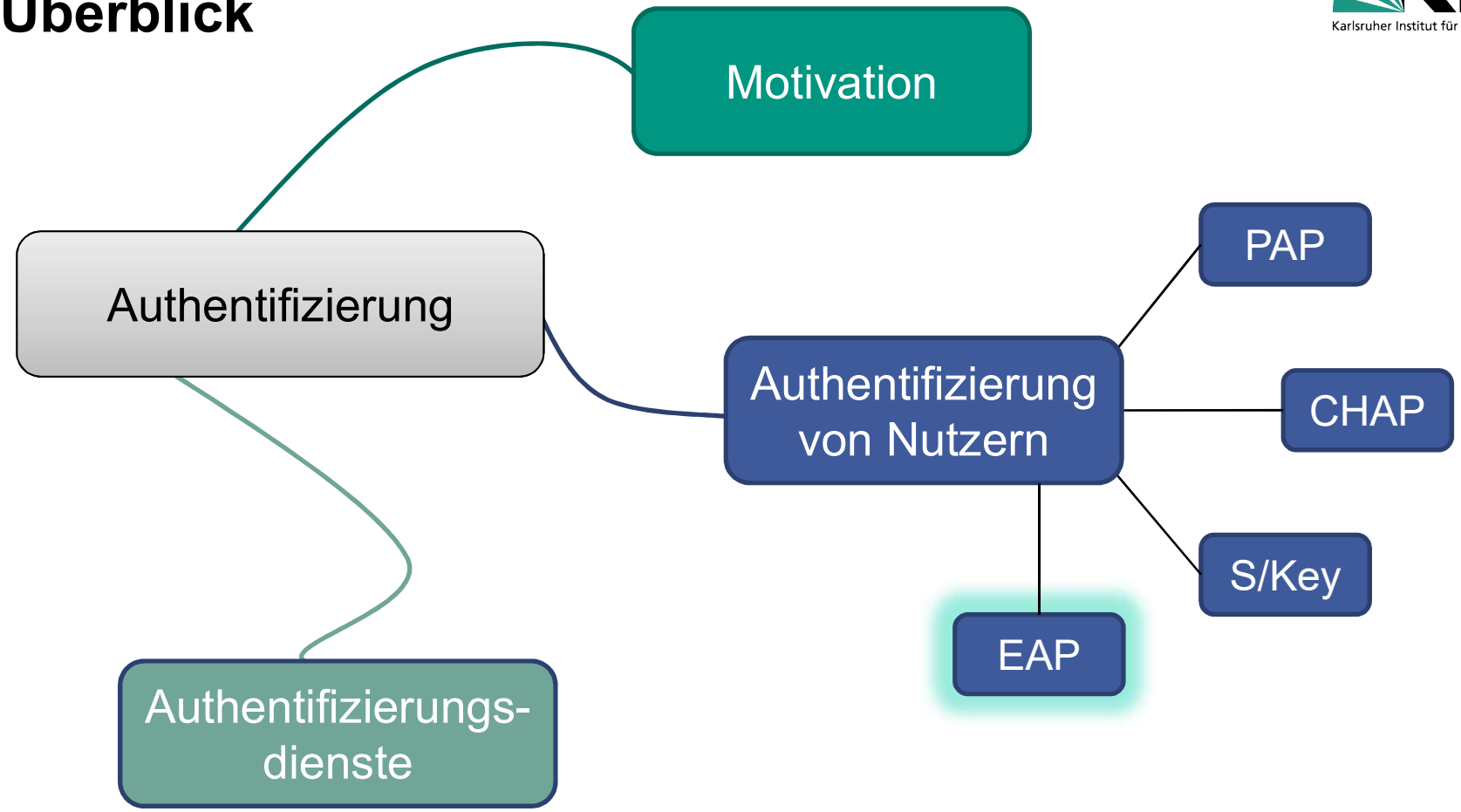
S/Key: Vorbereitung

- Client wählt
  - Password, Seed, Hash-Algorithmus
- Client berechnet Einmal-Passwörter S_n
  - $S_0 = \mathrm{H}(\mathrm{Password} + Seed)$
  - $S_1 = \mathrm{H}(S_0)$
  - ...
  - $S_n = \mathrm{H}(S_{n-1})$
  - Jeweils nur die ersten 64 Bit des Hash werden verwendet
- Client vereinbart „gesichert“ mit Ressource
  - Speichere das Paar $\{n, S_n\}$
- Client
  - Speichert alle Einmalpasswörter S_1 bis S_n

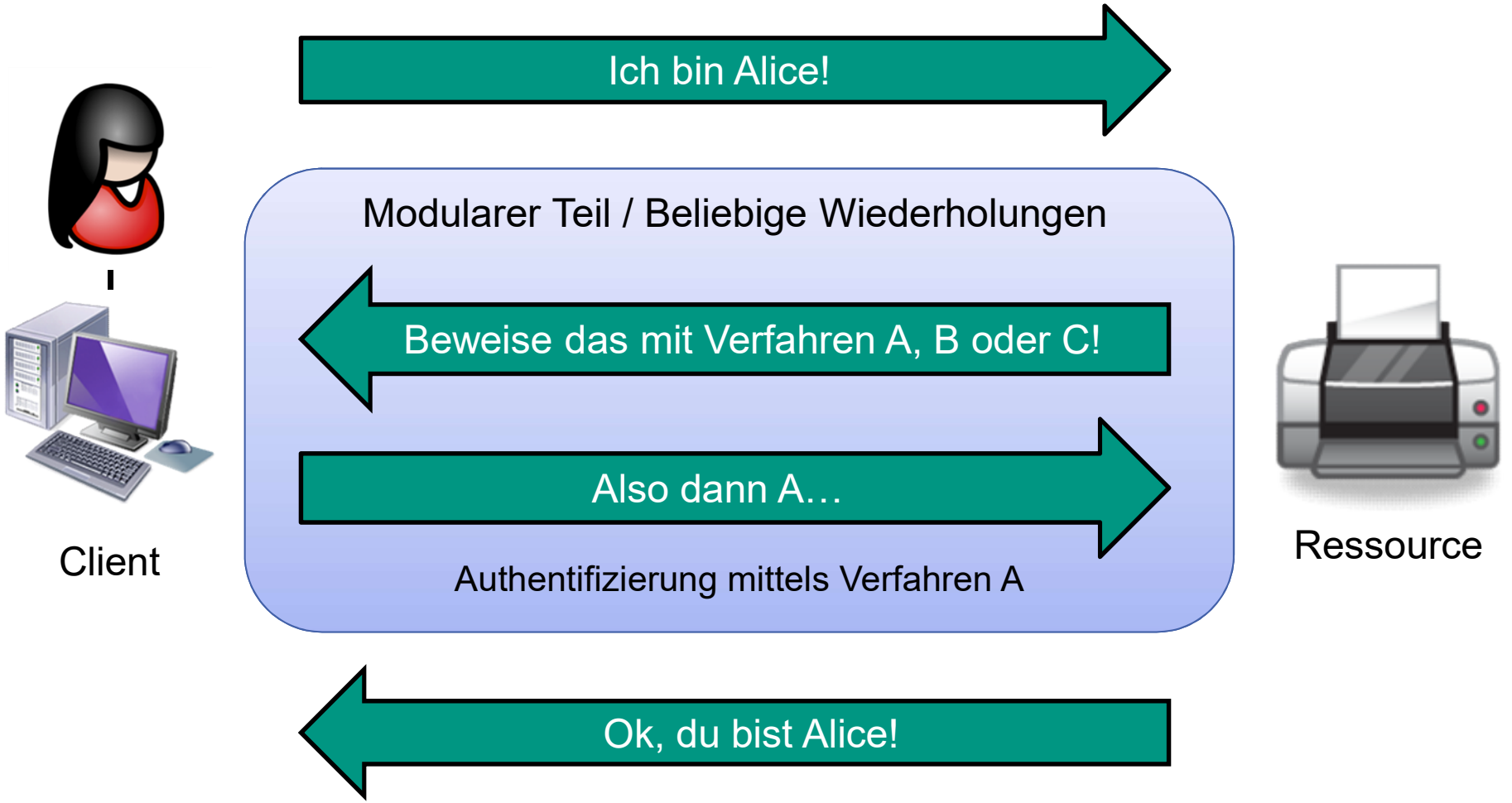
S/Key: Ablauf

- Ablauf der Authentifizierung
  - Client meldet sich mit Authentifizierungswunsch
  - Ressource schickt Challenge n an Client
  - Client sendet S_{n-1} zurück
  - Ressource
    - Verifiziert $\mathrm{H}(S_{n-1}) = S_n$
    - Schickt Bestätigung (oder Ablehnung) an Client
    - Löscht S_n
    - Speichert Paar $\{n-1, S_{n-1}\}$ für nächste Authentifizierung
  - Bevor letztes Passwort S_0 benutzt
    - Neue Initialisierung mit neuem Seed!
- Nächste Authentifizierung dann mit $n-1$

Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
PAP
CHAP
S/Key
EAP
Authentifizierungs-dienste

Modulare Authentifizierung



Modulare Authentifizierung

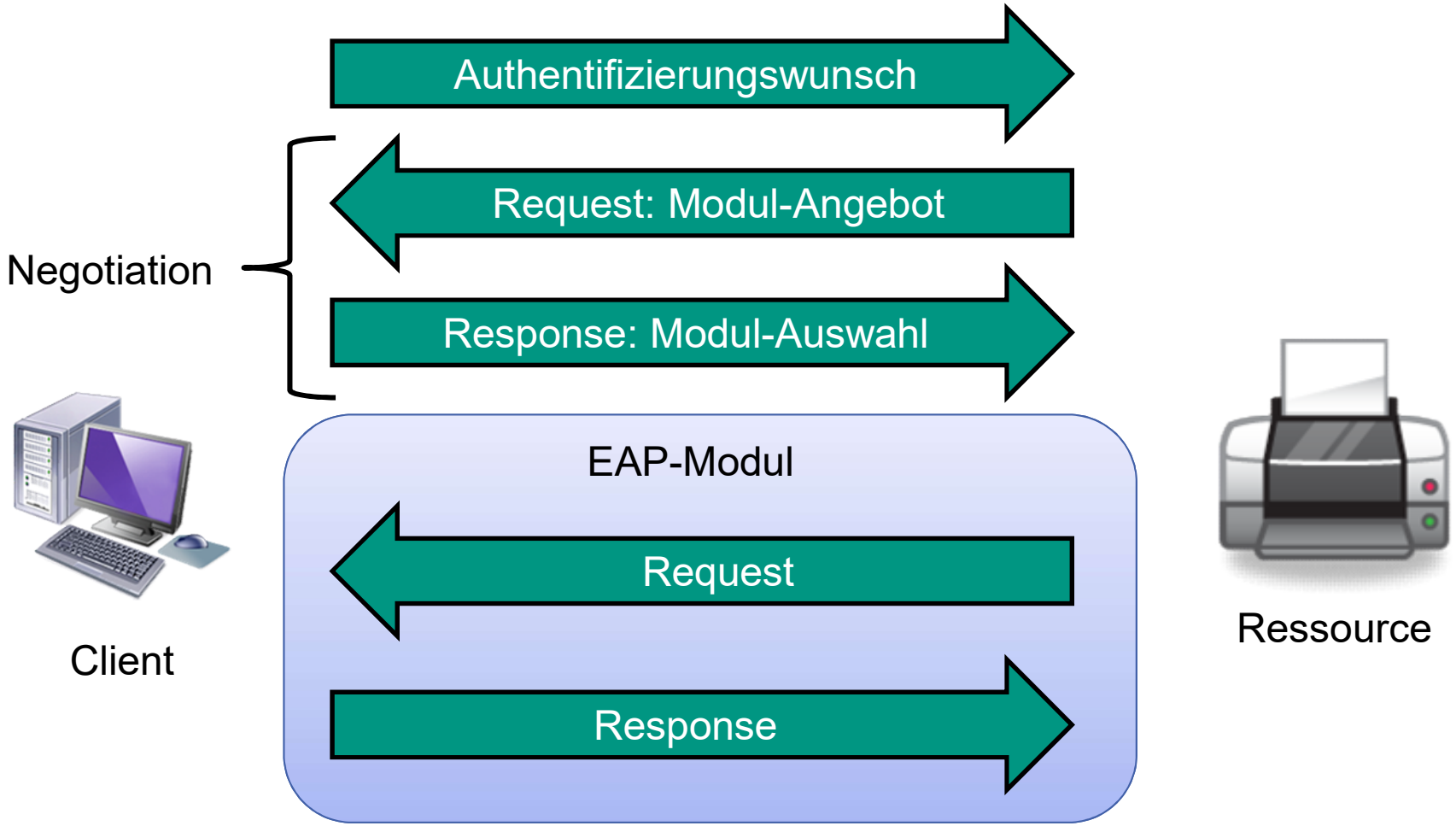
- Client äußert Authentifizierungswunsch
- Ressource sendet Antwort mit unterstützenden Verfahren
  - Liste der Verfahren beliebig erweiterbar
- Client wählt ein Verfahren aus
  - Oder unterbreitet Gegenvorschlag
- Authentifizierung wird mit gewähltem Verfahren ausgeführt
- Entscheidung, ob Authentifizierung erfolgreich ist oder nicht
  - Eventuell Wiederholung mit anderen Modulen

Extensible Authentication Protocol (EAP)

- Idee
  - Generisches Protokoll zur Authentifizierung
  - Unterstützung beliebiger Module zur Authentifizierung
- Ablauf
  - Client sendet Authentifizierungswunsch an Ressource
  - Ressource sendet eine / mehrere Anfragen an Client
    - z.B. Frage nach ID, Angebot von Modulen, Challenge, …
  - Client sendet angeforderte Daten zurück oder lehnt Anfrage ab
  - Ressource antwortet mit Erfolg / Misserfolg, wenn keine weiteren Fragen anstehen bzw. wenn Authentifizierungsergebnis feststeht
- Ablauf dynamisch anpassbar!

[RFC3748]

EAP: Nachrichtenaustausch



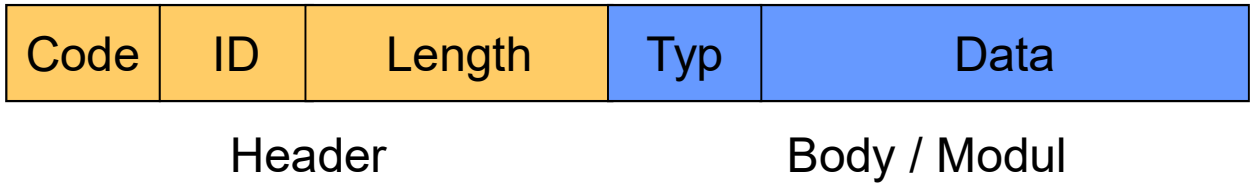
Entscheidung über Authentifizierung

EAP-Module

- Beispiele
  - MD5-Challenge
    - Ähnlich wie CHAP
  - Generic Token Card (GTC)
    - wie CHAP, nur Hardware-basiert
  - One-Time Password (OTP)
    - basierend auf Hash-Ketten
- Modulangebot lässt sich erweitern
  - Viele proprietäre Module spezifiziert
  - EAP-Modul Datenformat

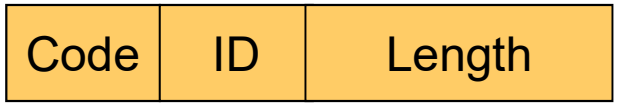
EAP-Dateneinheit

- Format für Request / Response



- EAP-Header
  - Code: EAP-Nachrichtentyp
    - Request, Response, Failure, Success
  - ID: Zuordnung von Antworten zu Anfragen
  - Length: Länge

- Format für Success / Failure



Verpflichtende EAP-Module

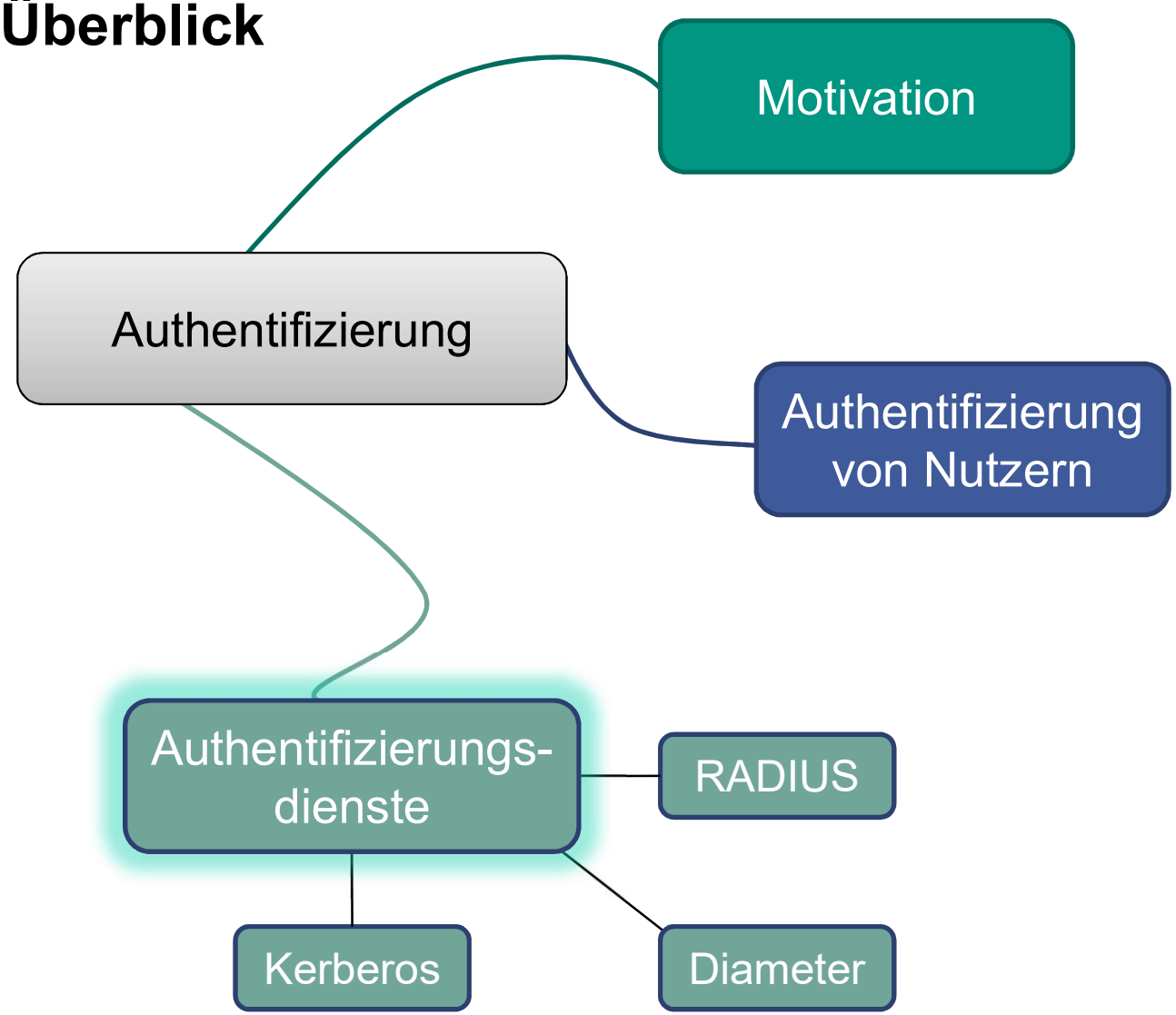
- Identity
  - Ermittlung der ID des zu authentifizierenden Clients
  - Meist erster Protokollaustausch
- Notification
  - Übertragung einer Nachricht an Client, die bestätigt werden muss
  - z.B. Warnung, dass Passwort bald erneuert werden muss
- NAK
  - Nur als Antwort verwendbar
  - Ablehnung einer Anfrage

Fazit zu Authentifizierungsmethoden

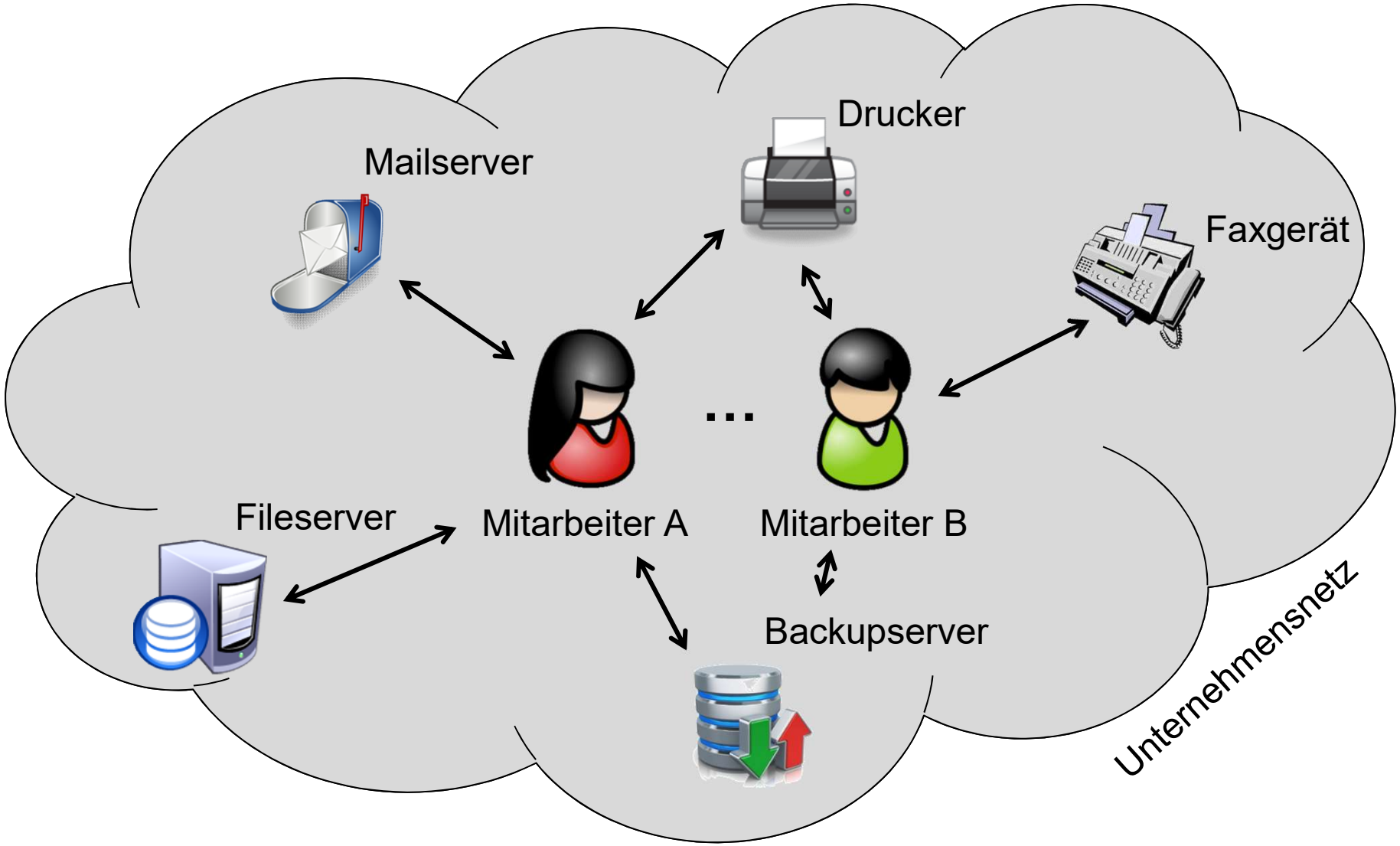
	Natives PAP	Gesichertes PAP	CHAP	MS-CHAP v2	S/Key	EAP
Abhören	✗	✓	✓	(✗)	✓	(✓)
Replay	✗	✓	✓	(✗)	✓	(✓)
Server	(✓)	(✓)	✗	✗	✓	(✓)
Baustein	Plain	Kanal	Hash+Nonce	Enc+Hash	Hash	Modul
Authent.	Client	Client+ Server	Client	Client+ Server	Client	Modul

- Natives PAP nicht sicher
- Bei CHAP liegen Passwörter im Klartext auf der Ressource
- EAP Sicherheit hängt von konkret verwendetem Modul ab
- Authentifizierung bietet nicht Vertraulichkeit/Integrität!
  - Zusätzlicher Schutz auf Transportweg notwendig
  - Insbesondere bei daran anschließender Kommunikation

Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
Authentifizierungs-dienste
RADIUS
Kerberos
Diameter

Beispiel: Im Unternehmen

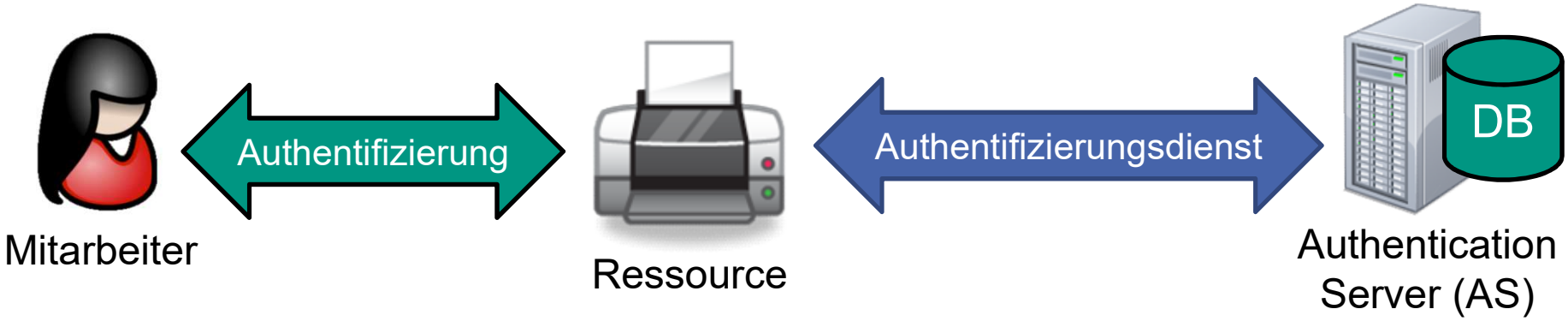


Wie kann der Betreiber Authentifizierung für mehrere Ressourcen auf einmal ermöglichen?

Authentifizierung bei mehreren Ressourcen

- Naiver Ansatz: „Copy&Paste“ der Nutzer-Datenbank
  - Sehr hoher Synchronisationsaufwand
  - Sicherheitsrisiko, Einbruch auf einer Ressource genügt
  - Zusätzlich: Autorisierung und Accounting kompliziert

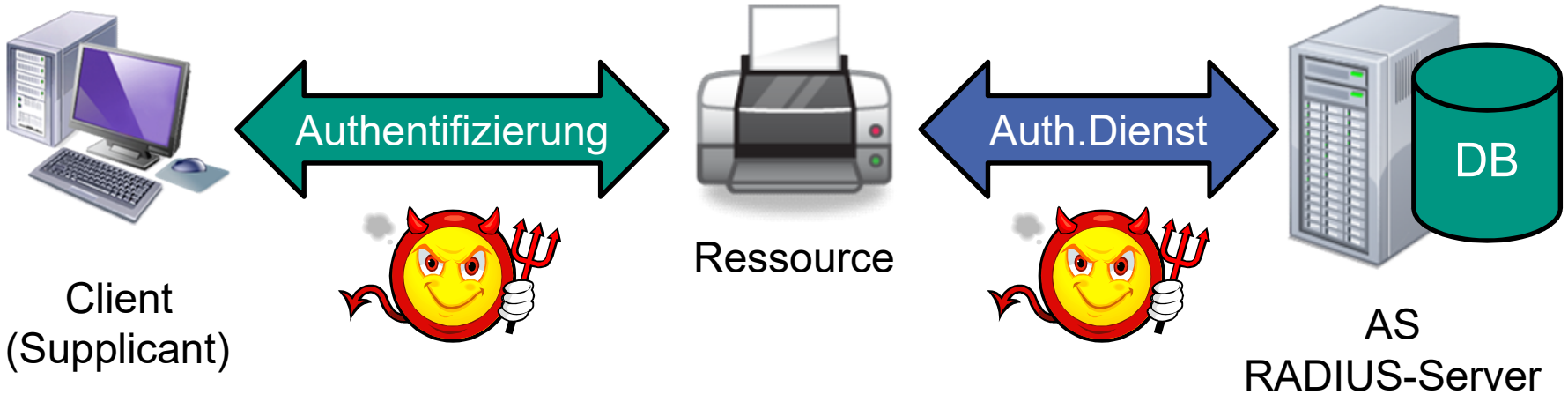
- Besser: Verwendung von Authentifizierungsdiensten
  - Auslagerung aller Nutzerdaten auf Authentication Server (AS)
  - Ressourcen leiten dann Authentifizierungsanfragen weiter



Authentifizierungsdienste

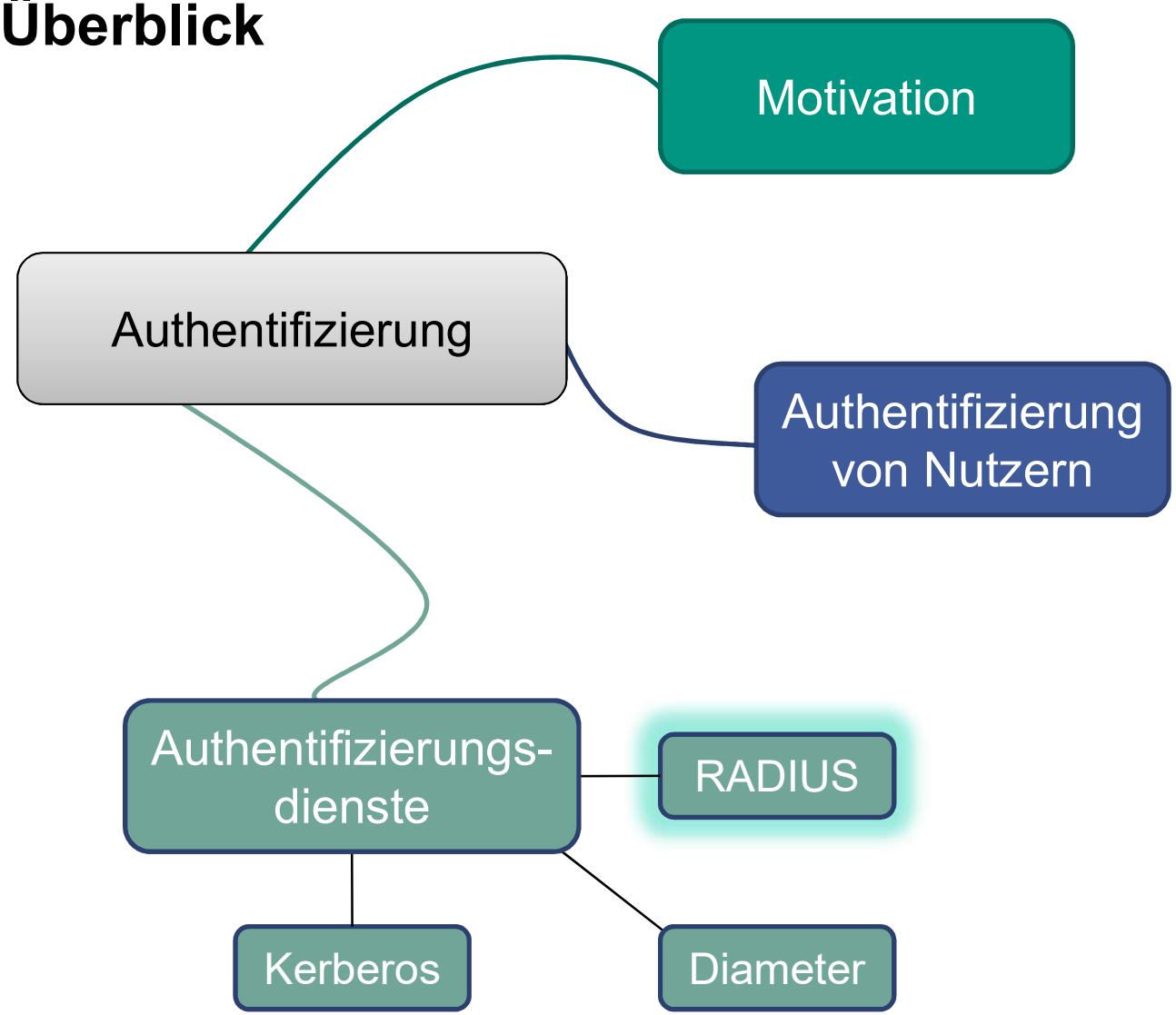
- Ziel: Auslagerung der Authentifizierung auf zentralen AS
- Typische Protokolle
  - Remote Authentication Dial In User Service (RADIUS)
  - Diameter – Nachfolger von RADIUS
  - Kerberos – Single Sign On (SSO)
- Zusätzliche Aufgaben
  - Wer? Was? Wie viel?
  - Protokoll zur Unterstützung von Triple-A (AAA)
  - Ausliefern von Rechten / Einstellungen für Nutzer

Angriffspunkte bei Authentifizierungsdiensten



- Bedrohung
  - Abhören, Einfügen, Verändern von Daten
- Authentifizierung (Client ↔ Ressource)
  - Absicherung wie gehabt – z.B. gesicherter Kanal
- Authentifizierungsdienst (Ressource ↔ AS)
  - Muss zusätzlich geschützt werden

Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
Authentifizierungs-dienste
RADIUS
Kerberos
Diameter

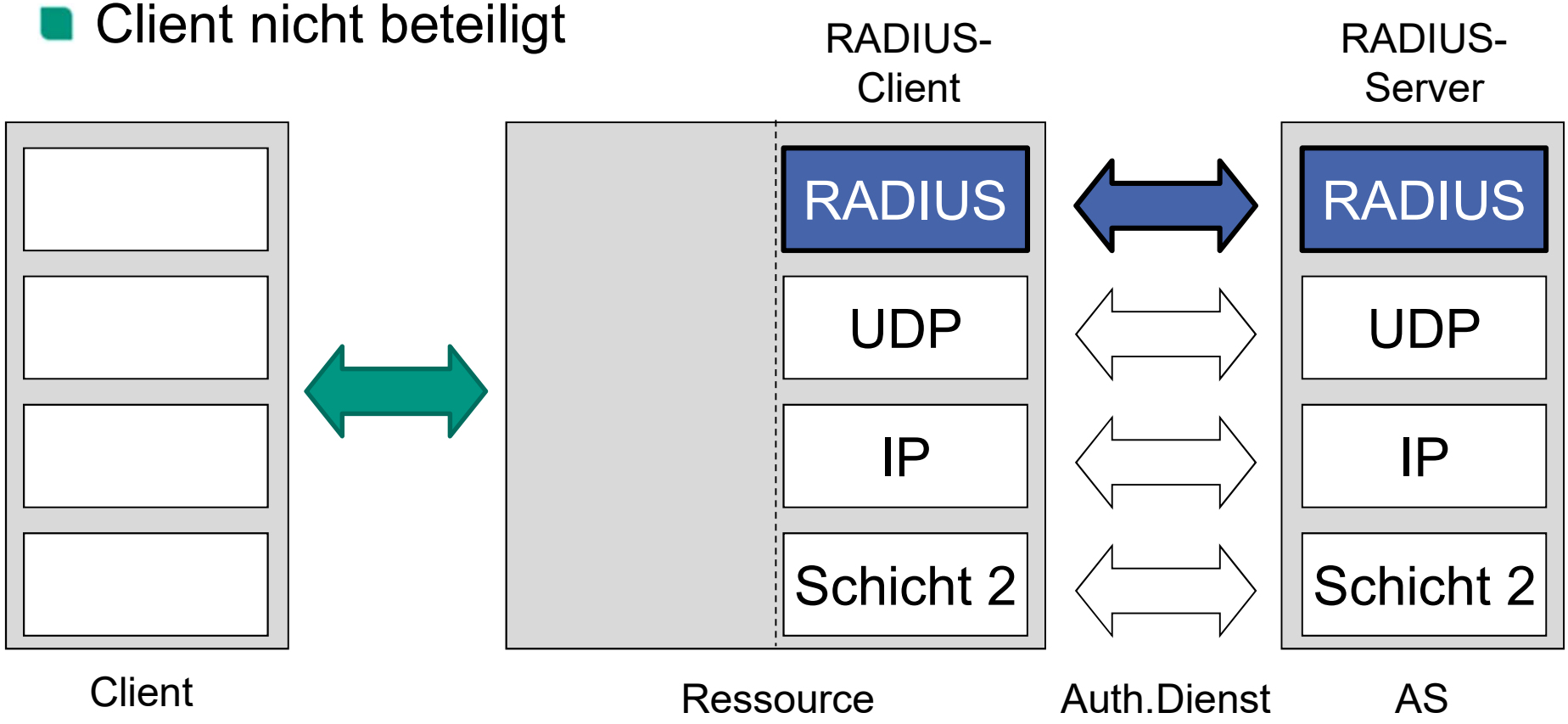
RADIUS

- Aufgabe
  - Transport von Authentifizierungsdaten (Ressource ↔ AS)
  - Proxyfunktion zum Weiterleiten an anderen AS (Roaming)
- Klassisches Client / Server Protokoll
  - Ressource ist Client
  - AS ist Server
- Ursprünglich nur Unterstützung von PAP und CHAP
  - Mittlerweile auch EAP

[RFC2865]

RADIUS im Protokollstapel

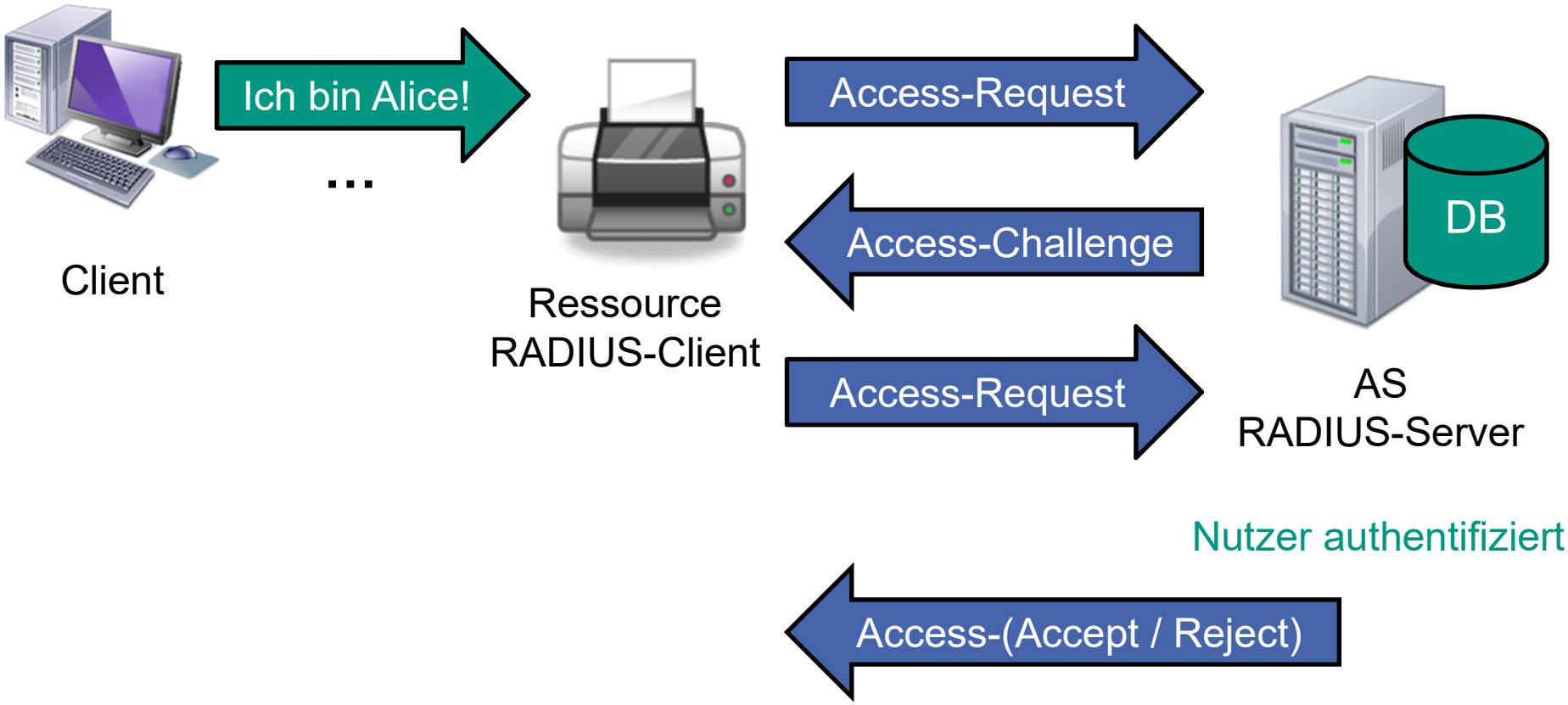
- Arbeitet in Anwendungsschicht über UDP
- Nur zwischen Ressource und AS
- Client nicht beteiligt



RADIUS: Rollen

- Vorbedingung
  - Client (Supplicant) stellt Konnektivität mit Ressource her, beginnt Authentifizierung
- Ressource: RADIUS-Client
  - Handelt Authentifizierungsverfahren mit Client aus
  - Übermittelt Authentifizierungsanfragen zu AS
  - Empfängt Authentifizierungsentscheidung von AS
- AS: RADIUS-Server
  - Nimmt Anfragen von Ressource entgegen
  - Authentifiziert und autorisiert Nutzer
  - Teilt RADIUS-Client Entscheidung mit
  - Oder leitet diese zu betreffendem AS weiter (Roaming)

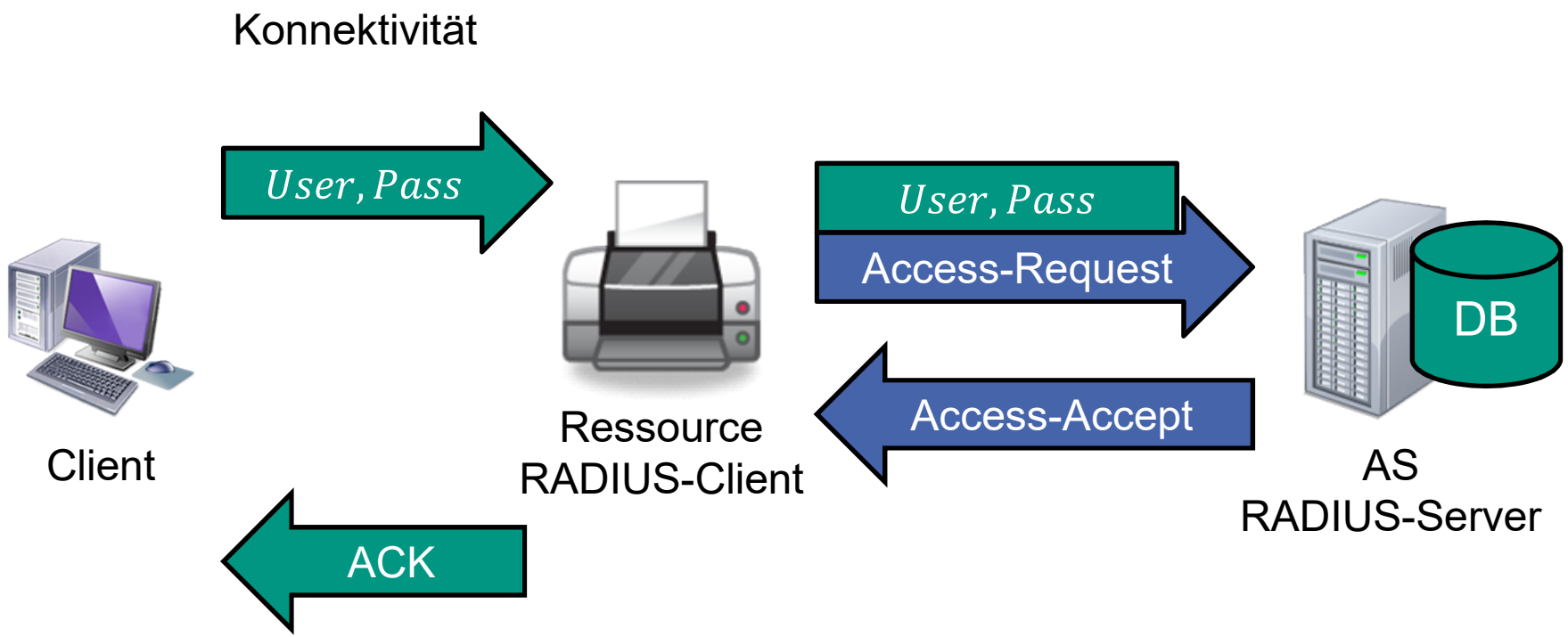
RADIUS: Protokollablauf



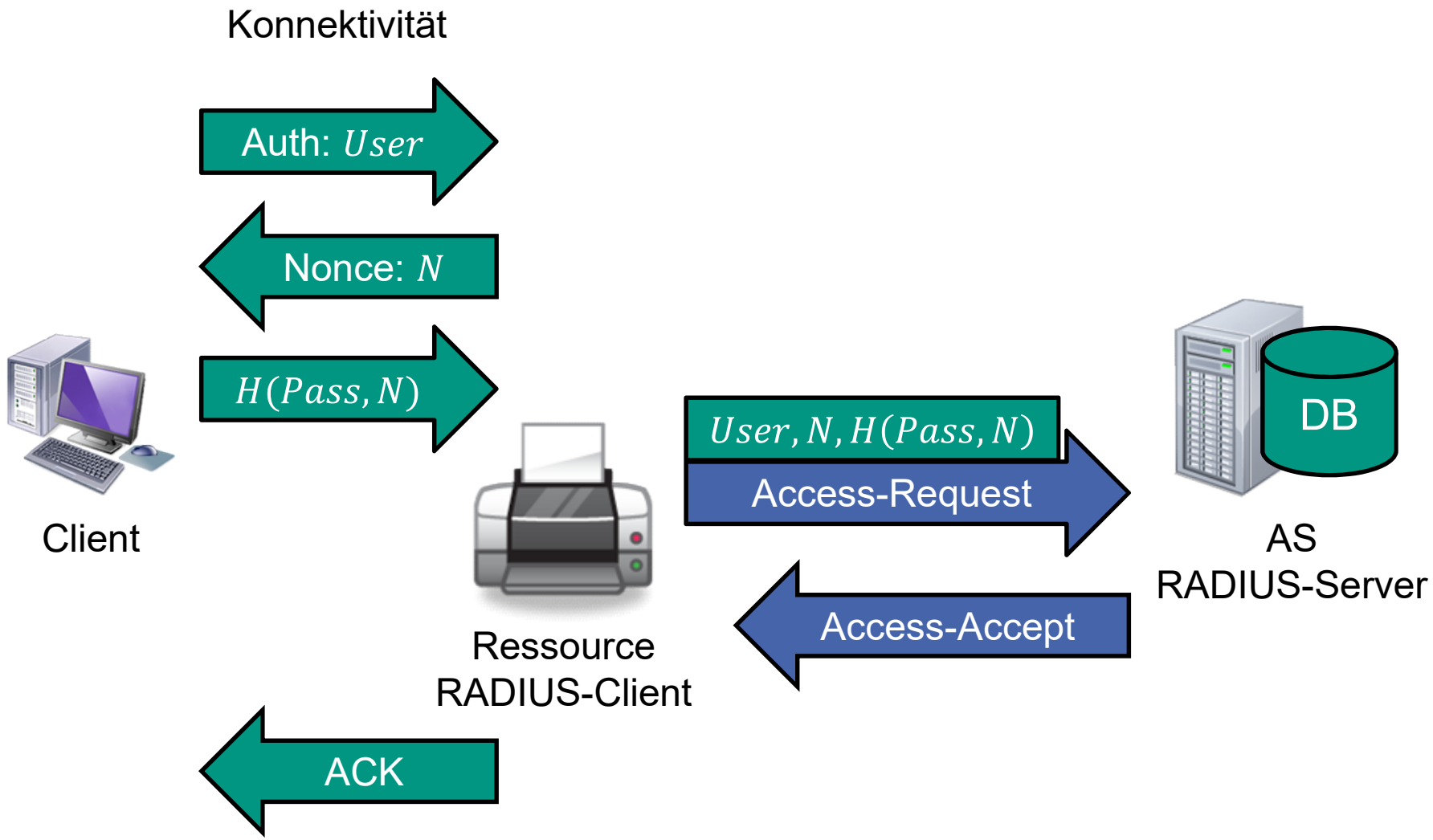
RADIUS: Nachrichtentypen

- Access-Request
  - Nachricht der Ressource zum AS
    - Weiterleitung der Anfrage des Client (Supplicant)
    - Antwort auf Anfrage (Challenge) des AS
- Access-Challenge
  - Nachricht des AS zur Ressource
    - Aushandlung der Authentifizierung und Autorisierung
- Access-Accept / Access-Reject
  - Nachricht des AS zur Ressource
    - Accept zum Bestätigen einer Authentifizierung
    - Reject zum Ablehnen eines Request

RADIUS: Authentifizierung mit PAP



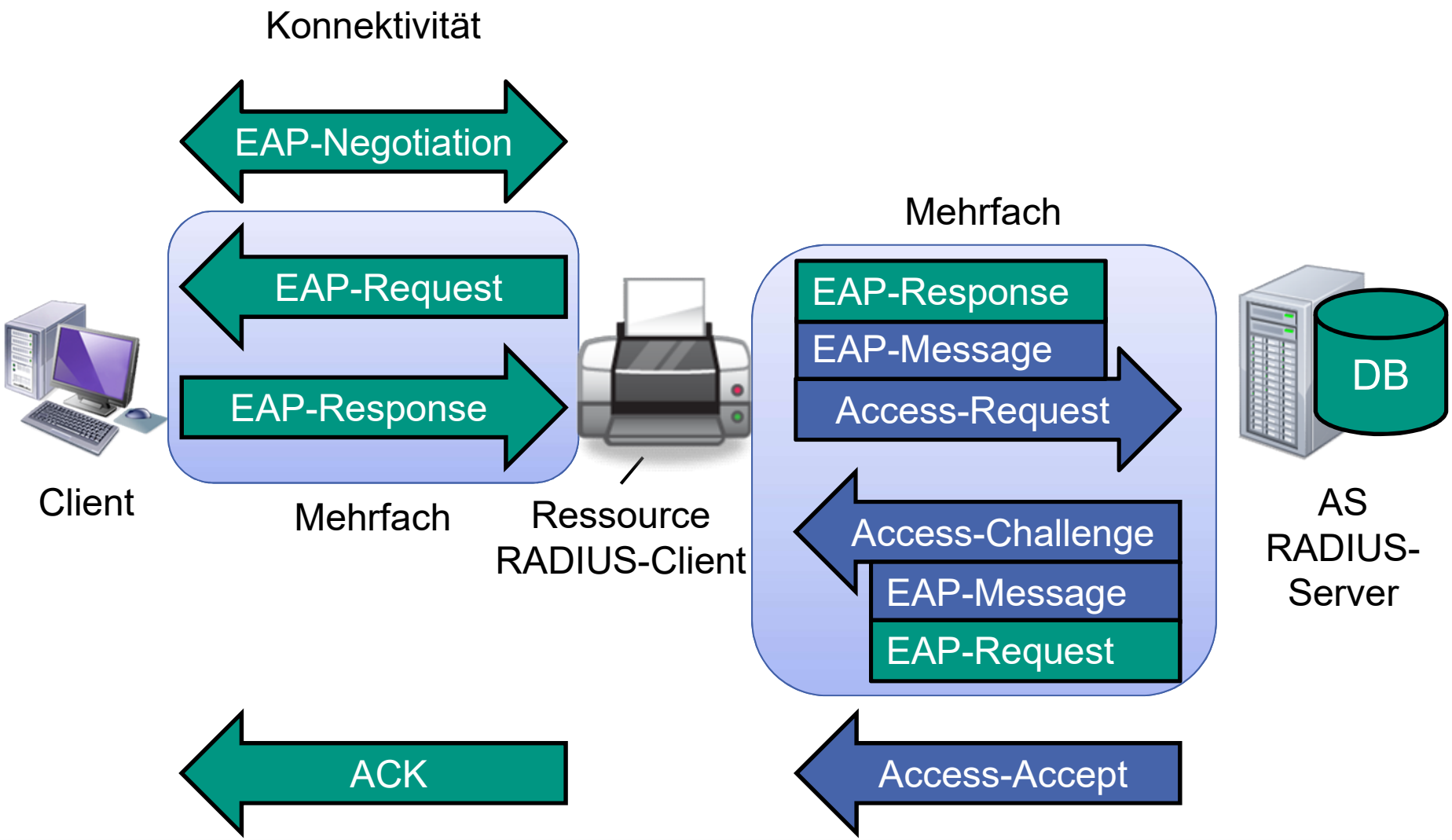
RADIUS: Authentifizierung mit CHAP



RADIUS: Authentifizierung mit EAP

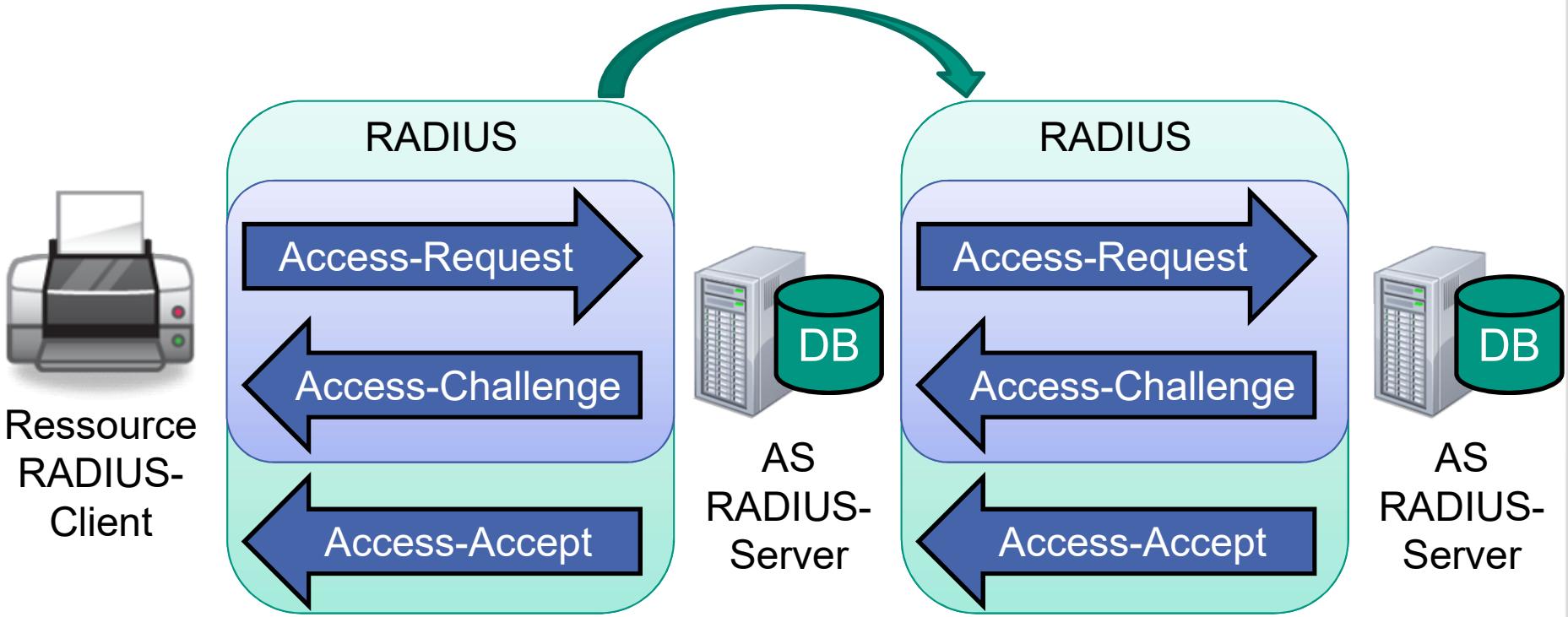
- Ressource authentifiziert Client meist über EAP
  - z B. bei PPP, PPPoE, 802.1X
  - Zwischen EAP-Paketen und RADIUS-Attributen müssten Daten aufwendig kopiert / umgesetzt werden
    - Kenntnis des genauen Formats für die verschiedenen Verfahren notwendig
- Idee
  - Unveränderter Transport der EAP-Pakete aus Frontend per RADIUS zwischen Ressource und AS (und zurück)
- RADIUS-Attribut: EAP-Message [RFC3579]
  - Direkter Transport von EAP-Paketen
  - Keine Übersetzung in einzelne RADIUS-Attribute notwendig

RADIUS: Authentifizierung mit EAP



RADIUS: Roaming

- Betrieb von mehreren RADIUS-Servern
  - Weiterleitung an anderen RADIUS-Server (Proxy)
  - Umkopieren von RADIUS-Nachrichten

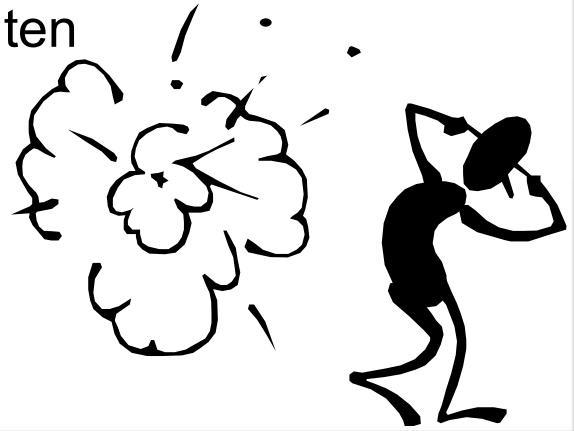


RADIUS: Sicherheitsbetrachtung

- Schutzziele (Ressource ↔ AS)
  - Vertraulichkeit der übertragenen Nachrichten
  - Integrität der übertragenen Nachrichten
  - Authentizität der Kommunikationspartner
- Umsetzung bei RADIUS
  - Gemeinsames Geheimnis zwischen Ressource und AS
    - RADIUS 16 Byte shared secret
  - Geheimnis wird gleichzeitig für Schutz der Integrität, Authentizität und Vertraulichkeit genutzt
    - Mittels der Hashfunktion MD5 (→ gebrochen!)
    - Request- / Response-Authenticator
    - Hidden Attributes

Fazit: RADIUS Sicherheitsmechanismen

- Rudimentärer Schutz durch RADIUS shared secret
  - Muss manuell verteilt werden
  - Fehlende Roaming-Unterstützung
  - Verwendet gebrochene Hashfunktion MD5
  - Schützt nicht alle Daten auf dem Kommunikationsweg
- Nur unzureichende Sicherheit vorhanden
  - Schwache Vertraulichkeit der Authentifizierungsdaten
  - Schwacher Schutz der Integrität von Nachrichten
  - Schwacher Schutz vor Replay-Angriffen



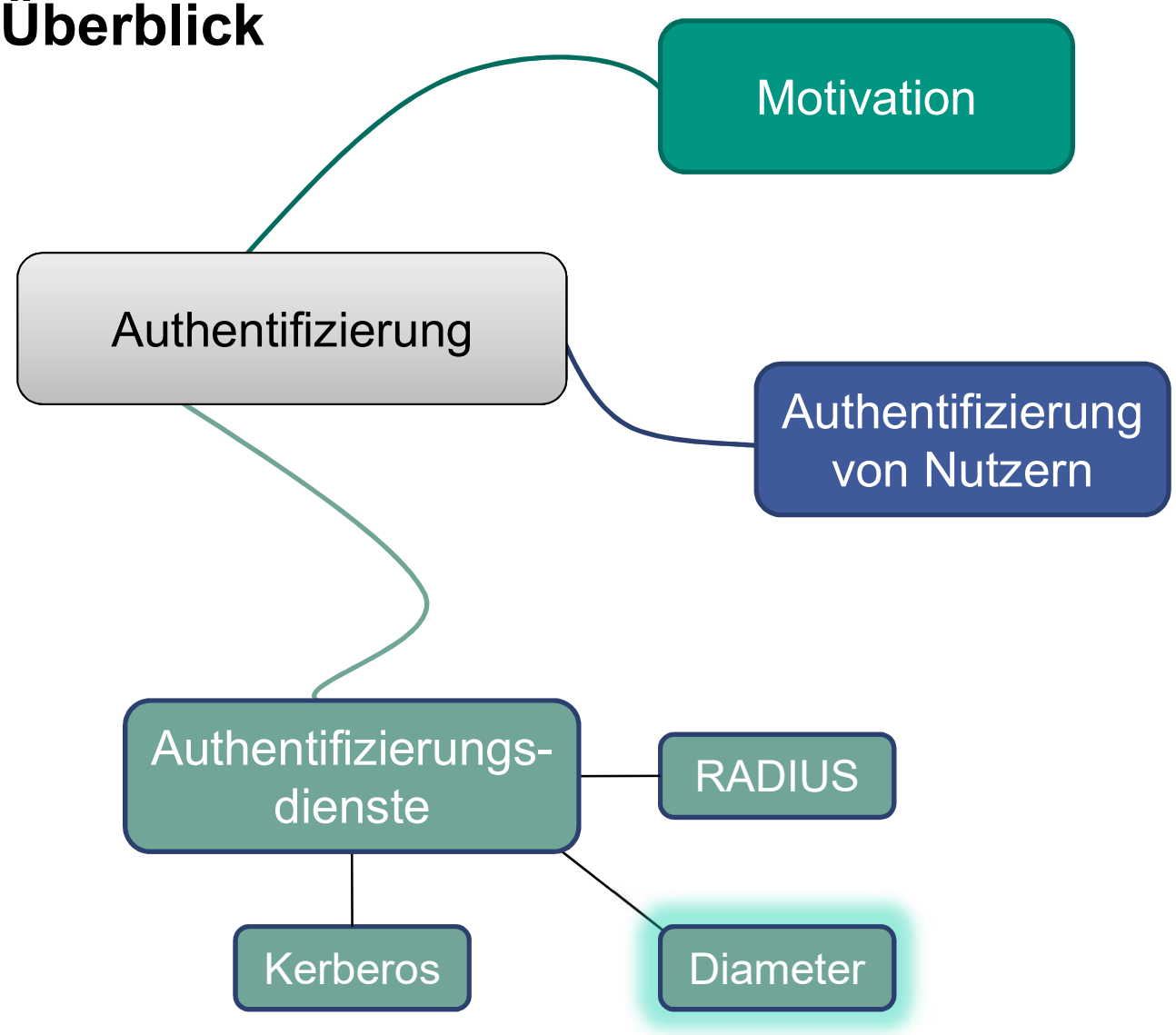
RADIUS over TLS

- RADIUS nutzt UDP zum Transport
  - Zwischen Ressource und AS
  - RADIUS muss sich selbst um Neuübertragung kümmern
  - Fragmentierung muss selbst von RADIUS gehandhabt werden
  - Keine Staukontrolle
  - Keine Absicherung der Kommunikation
- RFC6614: RADIUS auf Basis von TLS
  - Aufbauend auf TCP
  - Bietet Sicherheit auf Basis von TLS
    - Vertraulichkeit, Integrität und Authentizität

[RFC6614]

→ Eine Möglichkeit RADIUS sicher einzusetzen!

Überblick


Motivation
Authentifizierung
Authentifizierung von Nutzern
Authentifizierungs-dienste
RADIUS
Kerberos
Diameter

Diameter

- Nachfolger von RADIUS
  - Als Wortspiel: Twice the RADIUS
  - Nicht vollständig abwärtskompatibel zu RADIUS
- Vorteile gegenüber RADIUS
  - Verlässliche Transport-Protokolle (TCP oder SCTP)
  - Fokus auf Sicherheit
    - Verpflichtender Einsatz von IPsec oder TLS
  - Bessere Roaming-Unterstützung
  - Leichte Erweiterbarkeit
    - z.B. für neue Befehle und Attribute
  - Basisunterstützung für Accounting (Benutzersitzungen und Abrechnungen)

[RFC6733]

Diameter Application

- Diameter spezifiziert Framework für Anwendungen
  - Hohe Flexibilität und Erweiterbarkeit
  - z.B. Dienste, Protokolle, Mobile IP, Accounting
  - Aber genauso für Ressource und AS-Funktionalität
    - Zur Authentifizierung und Autorisierung

- Applikationen können Diameter-Funktionalitäten nutzen wie
  - Server-, Proxy-Komponenten
  - Verbindungsaufbau
  - Sitzungsmanagement
  - Kommunikationssicherheit

Diameter: Sicherheitsbetrachtung

- Von Anfang an Fokus auf Sicherheit
  - Verwendung von IPsec zwingend vorgeschrieben
    - Für Client oder Server Implementierung
  - Server muss zusätzlich TLS unterstützen
    - Client kann TLS unterstützen
- Hoher Schutz der Kommunikation
  - Zwischen Ressource und AS
- Kommunikation Client ↔ Ressource muss weiterhin zusätzlich gesichert werden
  - Gilt insbesondere für die sich an die Authentifizierung anschließende Kommunikation

Zusammenfassung

- Verschiedene „gute“ Authentifizierungsverfahren gegenüber passiven Angreifern
  - Über ungesicherten Kanal: OTP, S/Key
  - Über gesicherter Kanal (z.B. mit TLS): PAP, MS-CHAP v2
- Zum Schutz vor aktiven MitM-Angriffen: Immer zuerst Authentifizierung der Ressource erforderlich (z.B. über TLS)!
- Authentifizierungsdienste bisher
  - RADIUS, Diameter
  - Lösung zur Authentifizierung bei mehreren Ressourcen
  - Aber: am Besten auch über gesicherten Kanal
- Nächste Vorlesung: Kerberos
  - Authentifizierungsdienst für Single Sign On
  - Zusätzlich: Schlüsselaustausch, Autorisierung

Literatur

[RFC1334] B. Lloyd, W. Simpson; PPP Authentication Protocols; Oct. 1992

[RFC1760] N. Haller; The S/KEY One-Time Password System; Feb. 1995

[RFC1994] W. Simpson; PPP Challenge Handshake Authentication Protocol (CHAP); Aug. 1996

[RFC2289] N. Haller, C. Metz, P. Nesser, M. Straw; A One-Time Password System; Feb. 1998

[RFC2759] G. Zorn; Microsoft PPP CHAP Extensions, Version 2; Jan. 2000

[RFC2865] C. Rigney, S. Willens, A. Rubens, W. Simpson; Remote Authentication Dial In User Service (RADIUS); Jun. 2000

[RFC3539] B. Aboba, J. Wood; Authentication, Authorization and Accounting (AAA) Transport Profile; Jun. 2003

[RFC3579] B. Aboba, P. Calhoun; RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP); Sep. 2003

Literatur

[RFC3748] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, ; Extensible Authentication Protocol (EAP); Jun. 2004

[RFC6614] S. Winter, M. McCauley, S. Venaas, K. Wierenga; Transport Layer Security (TLS) Encryption for RADIUS; May 2012

[RFC6733] V. Fajardo, J. Arkko, J. Loughney, G. Zorn, Diameter Base Protocol; Oct. 2012