Naturwissenschaftlich Technische Fakultät
Department ETI

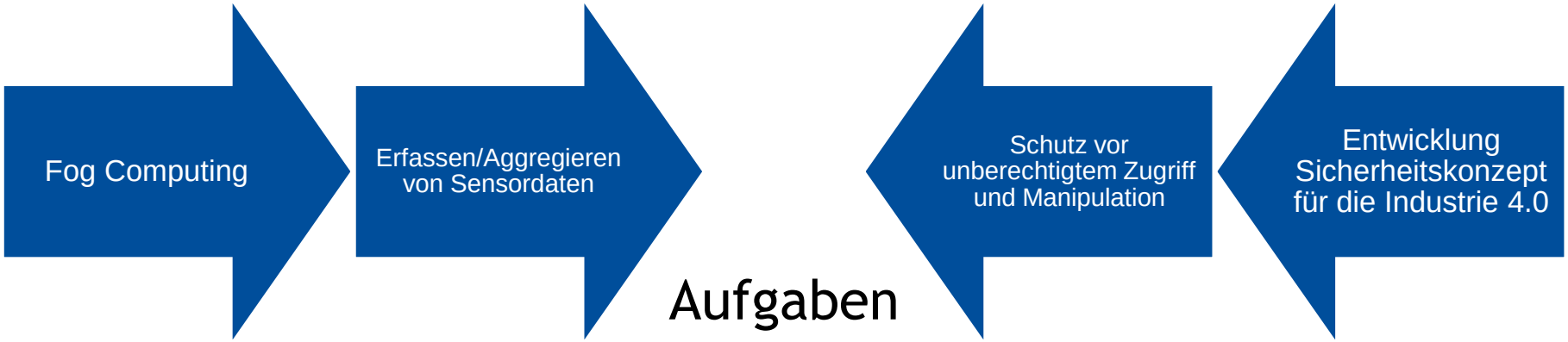
UNIVERSITÄT SIEGEN

Projektgruppe Sicherheit in Industrie 4.0 Fog Computing

Agenda

1. Anforderungen / Aufgabenstellung
2. Designprinzipen
3. Entwurf der Netz-Topologie
4. Sicherheitsaspekte
5. Softwarearchitektur
6. Prototyp
7. Fazit
8. Live-Demo

Anforderungen / Aufgabenstellung

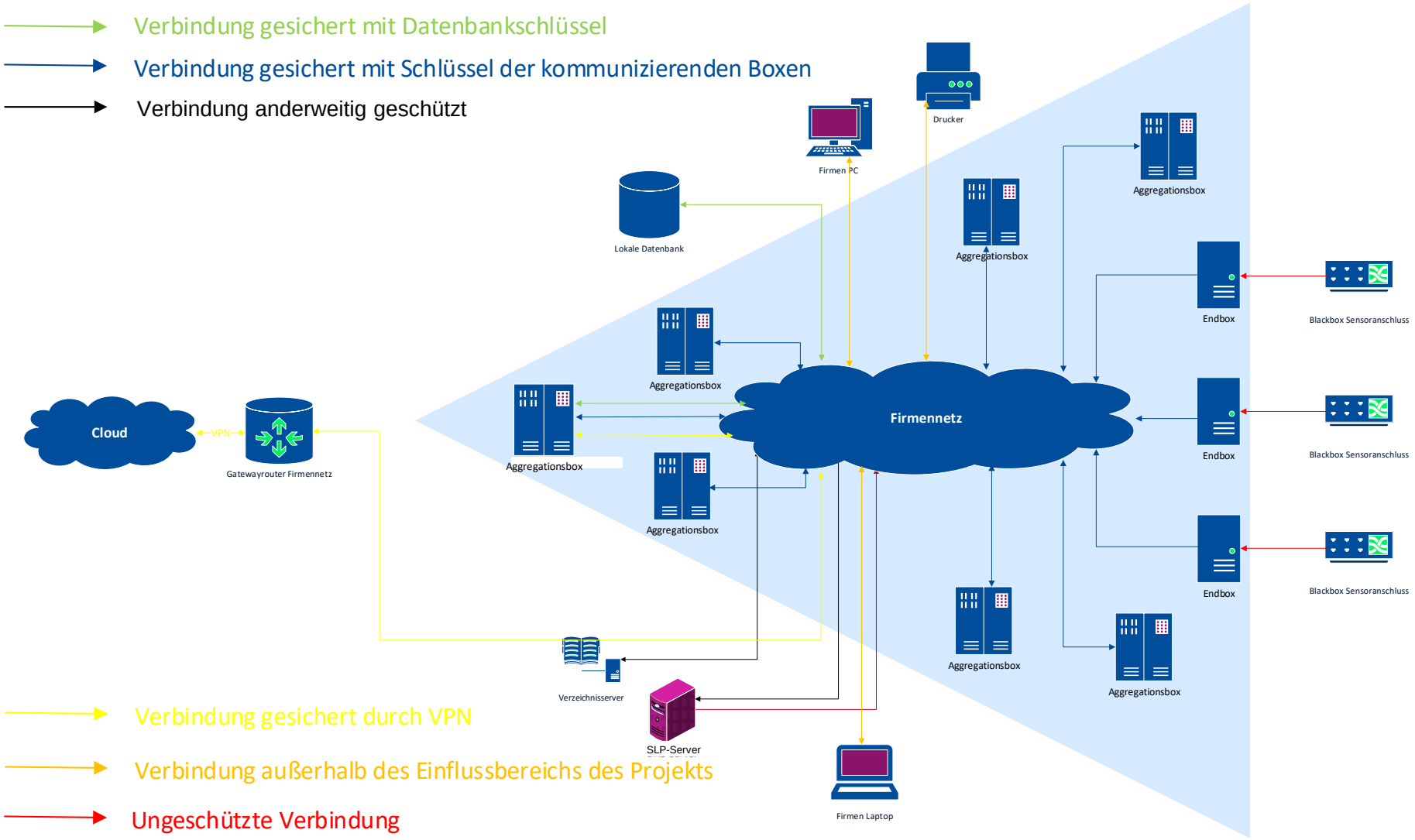


Aufgaben

- Entwurf der Netzwerkumgebung, Domänen, Rollen etc.
- Zugangskontrolle und Schlüsselmanagement
- Auswahl und Analyse der geeigneten kryptographischen Verfahren
- Implementierung von Ende-zu-Ende Sicherheitsmechanismen auf Anwendungsebene
- Bereitstellung einer Programmierschnittstelle

Designprinzipen

- Security by design
  - Sicherheit schon beim Netzwerk- und Softwareentwurf
- IoT-Gedanke
  - Vernetzung heterogener Hardware
- Plug&Play
  - Endgeräte nach Auslieferung und Aktivierung sofort einsatzbereit
- Skalierbarkeit
  - Innerhalb eines Standortes und standortübergreifend
- Erweiterbarkeit und Modularität
  - Einzelne Komponenten austauschbar


Verbindung gesichert mit Datenbankschlüssel
Verbindung gesichert mit Schlüssel der kommunizierenden Boxen
Verbindung anderweitig geschützt
Drucker
Firmen PC
Lokale Datenbank
Aggregationsbox
Aggregationsbox
Aggregationsbox
Endbox
Blackbox Sensoranschluss
Endbox
Blackbox Sensoranschluss
Endbox
Blackbox Sensoranschluss
Aggregationsbox
Aggregationsbox
Aggregationsbox
Aggregationsbox
Firmennetz
Cloud
VPN
Gatewayrouter Firmennetz
Verzeichnisserver
SLP-Server
Firmen Laptop
Verbindung gesichert durch VPN
Verbindung außerhalb des Einflussbereichs des Projekts
Ungeschützte Verbindung

Sicherheitsaspekte

- Sicherheitsdienste
- Crypto-Bibliothek
- Schlüsselmanagement
- Public-Key-Infrastruktur
- Zugriffskontrolle
- Remote-Entschlüsselung

Sicherheitsdienste im Einzelnen

- Vertraulichkeit
  - Schutz vor unbefugter Informationsgewinnung
- Authenticity
  - Umfasst **Integrität** und **Authentizität** des Ursprungs der Daten
- Nichtabstreitbarkeit
  - Empfänger kann sich sicher sein, dass Daten vom angegebenen Sender stammen und kann dies auch vor Dritten nachweisen
  - Stärker als Authenticity

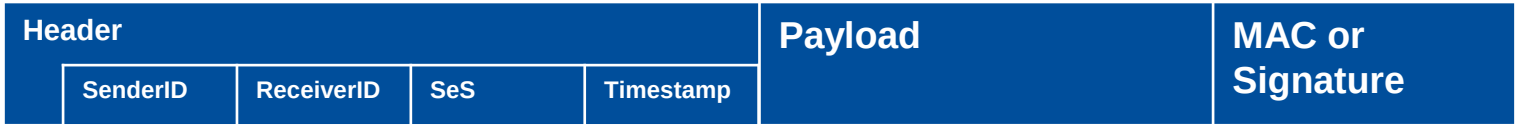
Kombinationen der Sicherheitsdienste

- Mögliche Kombinationen:
  0. Kein Security Service *(NONE)*
  1. Authenticity *(AUTH)*
  2. Vertraulichkeit + Authenticity *(CONF_AUTH)*
  3. Nichtabstreitbarkeit *(NONREP)*

- Zusätzlich Erkennung von
  - Reihenfolgevertauschung von Nachrichten
  - Replay von Nachrichten
  - Nachrichtenunterdrückung

Umsetzung der Sicherheitsdienste

- Kommunikationsprotokoll auf Anwendungsebene
  - Überträgt Security Service
  - Beinhaltet TVPs, Sequenznummern und ACKs
    - Erkennung von Reihenfolgevertauschung, Replay und Nachrichtenunterdrückung
  - Beispielnachricht:



- Durchführung der Sicherheitsmechanismen durch Crypto Libary

Crypto Library

- Realisierung der Sicherheitsdienste
  - *AUTH* und *CONF_AUTH* mit Hilfe standardisierter symmetrischer kryptografischer Verfahren
  - *NONREP* mit Hilfe standardisierter asymmetrischer kryptographischer Verfahren
- Verwaltung der symmetrischen Schlüssel
- Konzept:
  - Verwendet intern OpenSSL-Bibliothek → Wrapper in C++
  - Abstrahiert von kryptographischen Details z.B. Verwaltung von IVs
    - Einfache Benutzung
    - Fehlerquellen vermeiden
    - Rechtzeitige Schlüsselwechsel

Crypto Library

- Verfügbare Services:
  - Service AUTH: GMAC und HMAC (basierend auf SHA-256)
  - Service CONF_AUTH: GCM und CTR mit HMAC (parallelisiert)
  - Service NONREP: ECDSA mit Kurve NIST p-256 und SHA-256
- GCM und GMAC effizient auf „starken“ Prozessoren
  - Profitieren vom AES-NI Befehlssatz
- Counter-mode effizient auf kleinen Prozessoren mit vielen Kernen
  - Parallelisiert implementiert, bis zu 8 Threads
  - BananaPi M3: Octa-Core Cortex A7 (im Idealfall voll ausgelastet)

Crypto Library – Performancebeispiel

Algorithmus: GMAC

Intel Core i3 2120, 3,3GHz	1487 MB/s
Intel Core i5 2430M, 2,4GHz	1283 MB/s
Intel Core i3 530, 2,9GHz	354 MB/s
Quad-Core Cortex A8 (Rasp. Pi 3)	48.3 MB/s
Octa-Core Cortex A7 (Banana Pi)	36.9 MB/s
Quad-Core Cortex A7 (Rasp. Pi 2)	26.1 MB/s

Algorithmus: SHA-256

Intel Core i3 2120, 3,3GHz	245 MB/s
Intel Core i5 2430M, 2,4GHz	219 MB/s
Intel Core i3 530, 2,9GHz	177 MB/s
Quad-Core Cortex A8 (Rasp. Pi 3)	64.3 MB/s
Octa-Core Cortex A7 (Banana Pi)	36.5 MB/s
Quad-Core Cortex A7 (Rasp. Pi 2)	25.2 MB/s

Service „AUTH“

Algorithmus: GCM

Intel Core i3 2120, 3,3GHz	213 MB/s
Intel Core i5 2430M, 2,4GHz	716 MB/s
Intel Core i3 530, 2,9GHz	179 MB/s
Quad-Core Cortex A8 (Rasp. Pi 3)	22.6 MB/s
Octa-Core Cortex A7 (Banana Pi)	14.9 MB/s
Quad-Core Cortex A7 (Rasp. Pi 2)	11.1 MB/s

AES-NI

Algorithmus: Encrypt-then-MAC

Intel Core i3 2120, 3,3GHz	120 MB/s
Intel Core i5 2430M, 2,4GHz	186 MB/s
Intel Core i3 530, 2,9GHz	140 MB/s
Quad-Core Cortex A8 (Rasp. Pi 3)	42.7 MB/s
Octa-Core Cortex A7 (Banana Pi)	26.9 MB/s
Quad-Core Cortex A7 (Rasp. Pi 2)	18.4 MB/s

Service „CONF_AUTH“

Crypto Server

- Kommunikation zwischen CryptoClient und CryptoServer über RabbitMQ
  - Kommunikation ausschließlich lokal
  - Definition der Request/Response Datenpakete mithilfe von Google ProtocolBuffers
- Bindeglied zwischen CryptoLibrary und anderen Anwendungen
  - Ebenfalls in C++ geschrieben
  - Rest des Frameworks in Java implementiert
- Zusätzliche Features:
  - Verwaltung asymmetrischer Schlüssel

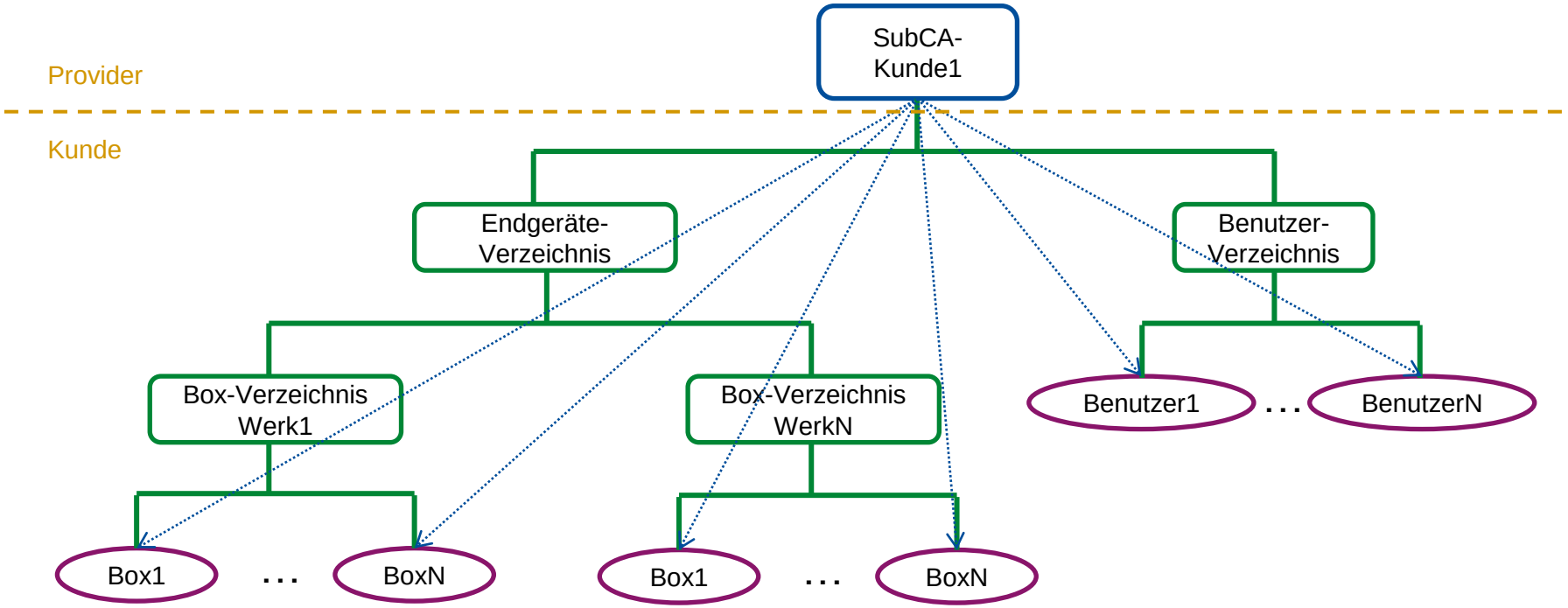
Public Key Infrastruktur

- Authentischer Schlüsselaustausch
- Sicherheitsdienst Nichtabstreitbarkeit
- Zugriffskontrolle
- TLS
- Andere Protokolle, die Zertifikate benutzen können

Nutzung des Verzeichnisdienstes durch PKI

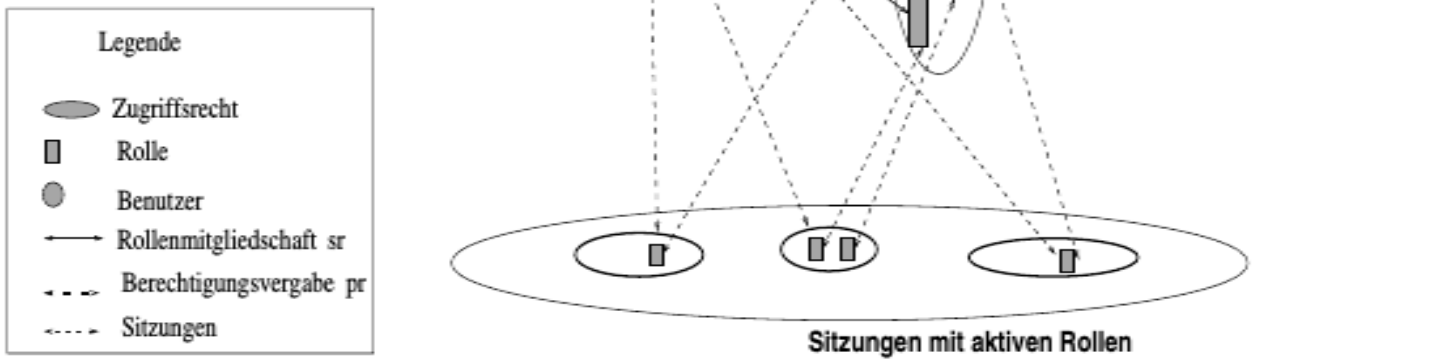
- Der Verzeichnisdienst wird benutzt, um die Zertifikate für alle (Boxen und Nutzer) zugänglich zu machen
  - Zertifikate abgespeichert, möglichst nahe am Besitzer
  - Zertifikate und Attribute (z.B. Widerrufsstatus) können angefragt werden
- Realisierung mit OpenLDAP
  - Lightweight Directory Access Protocol
  - Abbildung der Objekte auf baumartige Verzeichnisstruktur
  - Kompatibilität durch X.500-Modell des Verzeichnisdienst und X.509v3 Standard der Zertifikate

Topologie des Verzeichnisdienstes und der PKI



Zugriffkontrolle - Modell

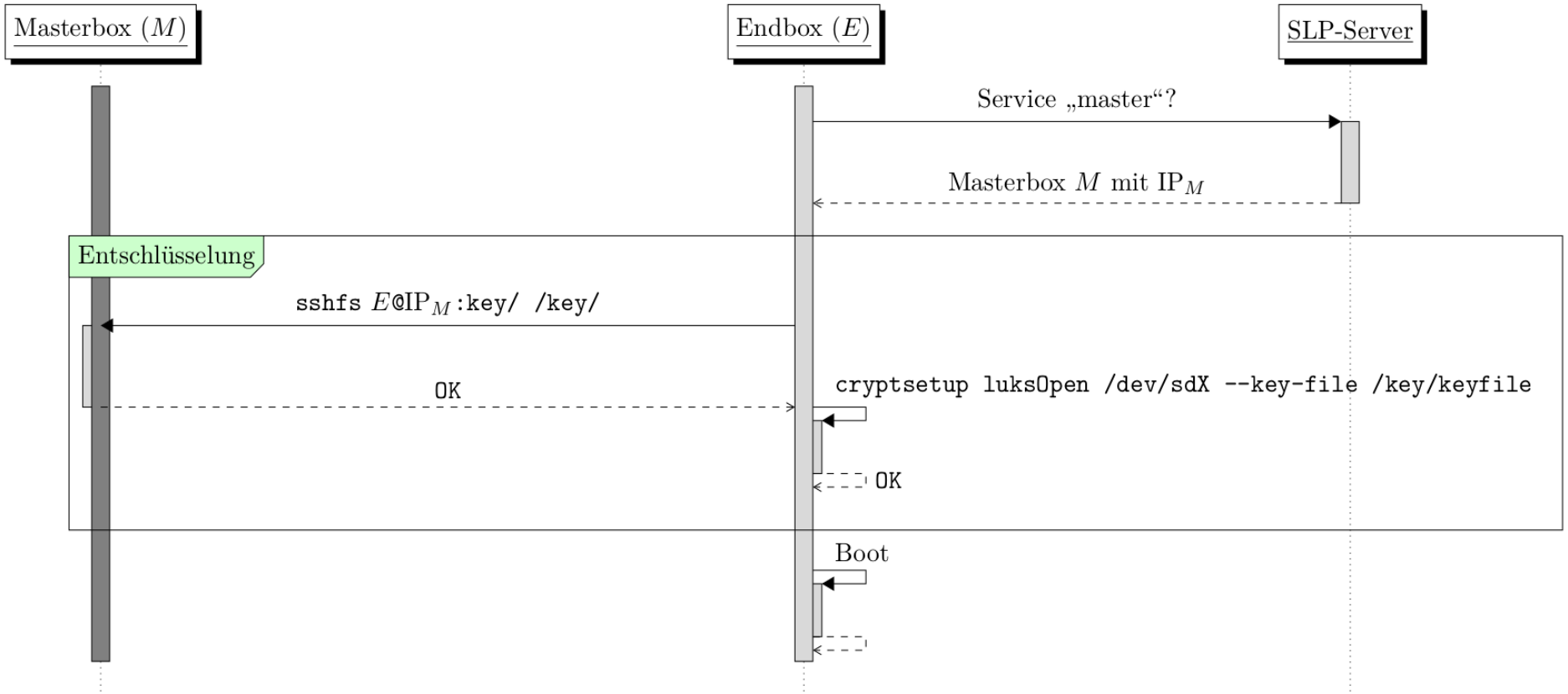
- Verwendung Rollenbasierter Zugriffskontrolle
- Rechtevergabe an aufgabenbezogene Rollen
- Zuordnung der Subjekte zu den Rollen in Sitzungen



Zugriffkontrolle - Technologien

- Kontrolle auf LDAP-DIT: über ACLs
  - In OpenLDAP bereits vorhanden
- Kontrolle im Netzwerk: über OpenRBAC
  - Kombination von RBAC mit LDAP
    - rbacRole: Objektklasse für Rollen
    - rbacRessource: Objektklasse für Objekte
    - rbacPerformer: Kennzeichnet Teilnehmer der Rolle
    - rbacOperation: Kennzeichnet durchzuführende Operation
    - Berechtigung: <Rollen-DN>:=:<Operation>
  - Prüfung über LdapProxy

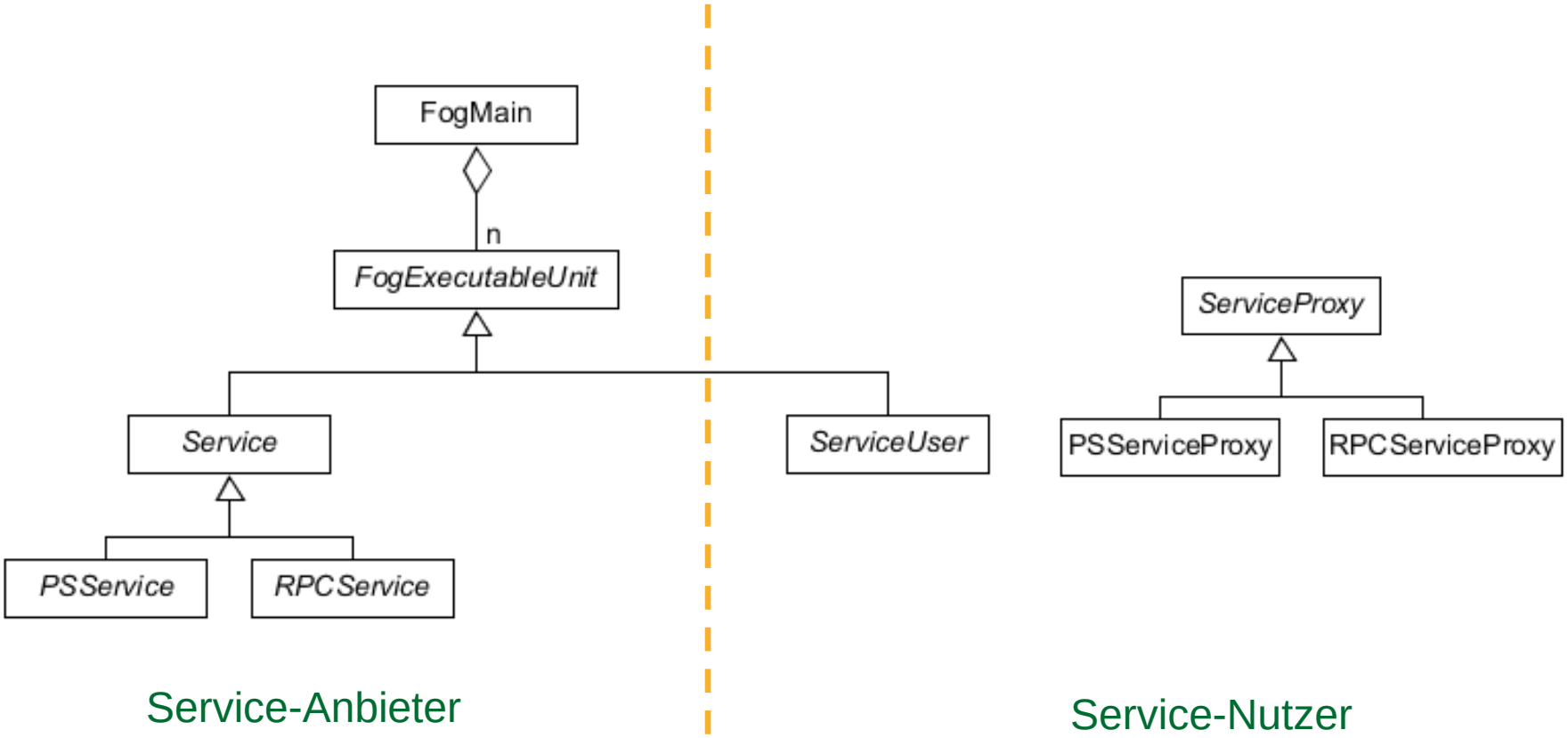
Remote-Entschlüsselung



Software-Architektur

- Serviceorientierter Entwurf
  - Es gibt Boxen, die Services bereitstellen
  - Es gibt Boxen, die Services nutzen
- Prinzipien:
  - Publish-Subscribe: Eine Box stellt periodisch Daten zur Verfügung, (mehrere) andere Boxen können sich registrieren und empfangen diese Daten
  - Remote Procedure Call: Eine Box stellt eine Prozedur zur Verfügung, eine andere Box kann diese Prozedur nutzen und erhält das Ergebnis per Nachricht

Software-Architektur



Software-Architektur

- RabbitMQ als nachrichtenorientierte Middleware
- Wird gekapselt in Java Framework
  - Stellt Schnittstelle nach außen bereit
  - Transparente Nutzung
- Transparente Umsetzung der Sicherheitsdienste
  - Security Service muss nur ausgewählt werden
- Entwickler können ihre eigenen Services programmieren

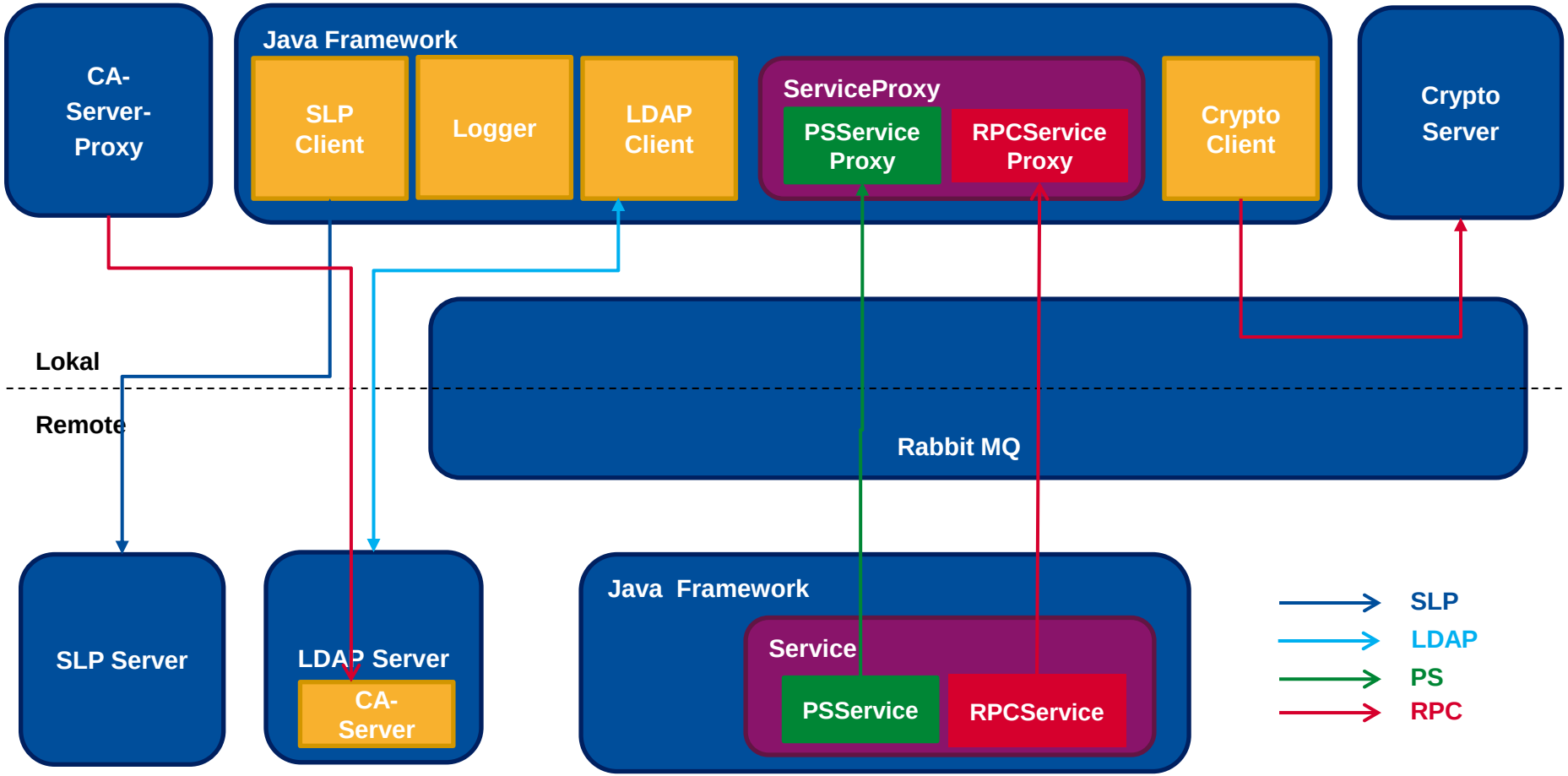
RabbitMQ

- Nachrichtenorientierte Middleware
  - In „Erlang“ implementiert
  - Implementiert das „Advanced Message Queuing Protocol“ (AMQP)
  - Client-Bibliotheken für Java verfügbar
- Unterstützte Kommunikationsmodelle
  - Remote Procedure Call (RPC)
  - Publisher-Subscriber (PS)

RabbitMQ

- Funktionsweise
  - Message-Broker verwalten Message-Queues
  - Nachrichten können in Queues geschrieben werden
  - Clients können Nachrichten aus Queues lesen
- Zugriff auf Queues durch Benutzername/Passwort regelbar
- Message-Broker und Client/Server können auf verschiedenen Knoten im Netz laufen
- MessageQueues können dynamisch erstellt werden
  - Zugriff auf Queues über den Namen der Queue

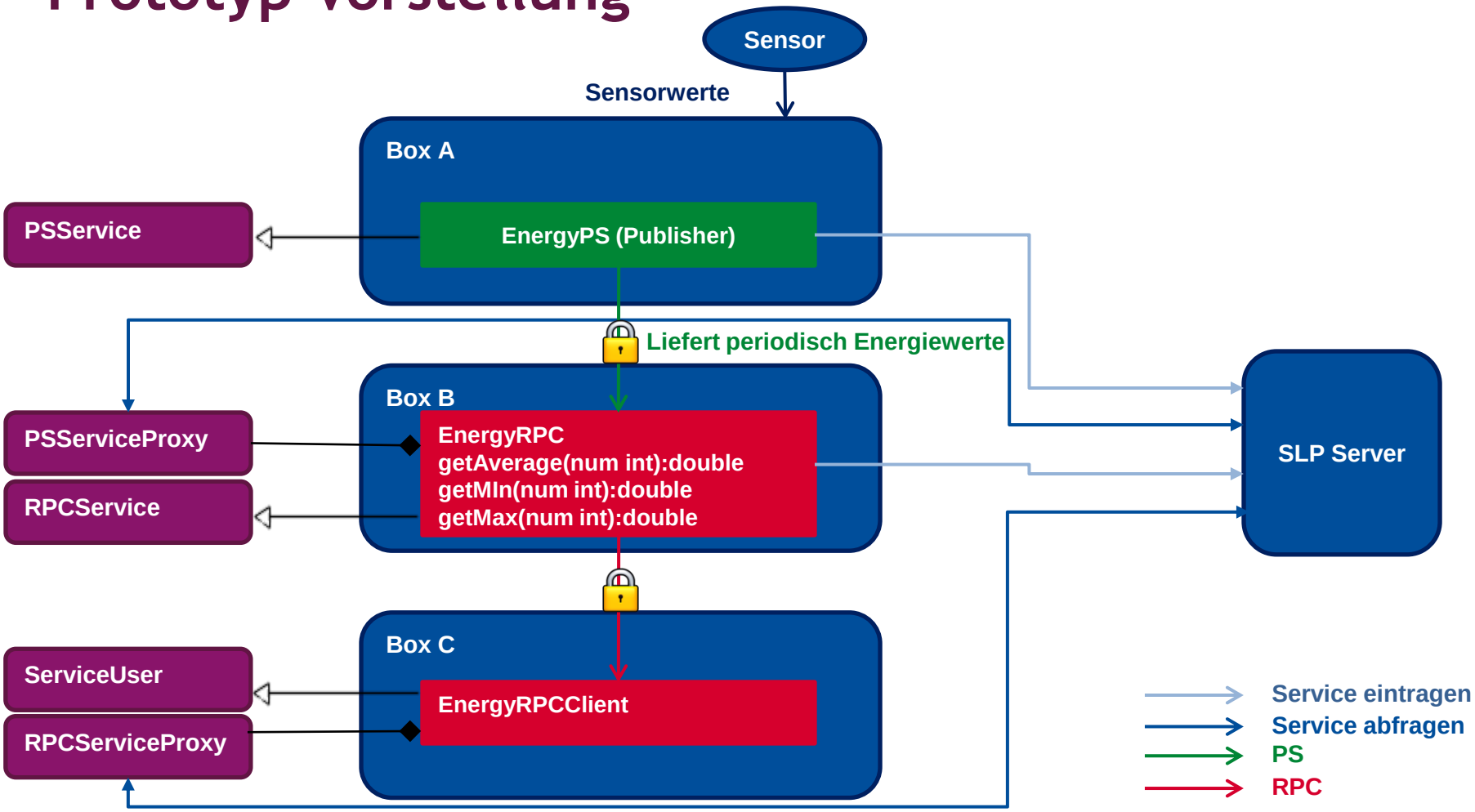
Software-Architektur



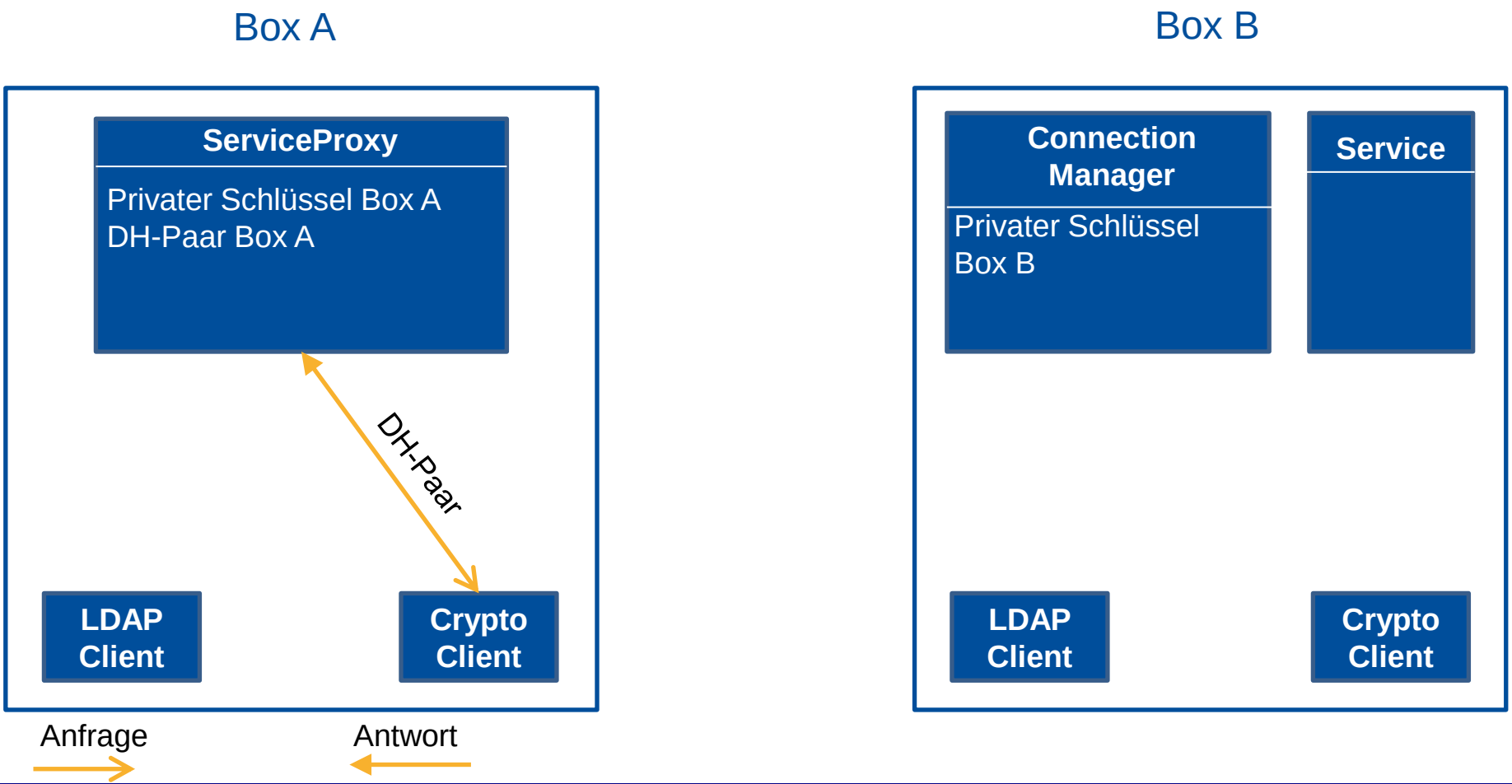
Prototyp Vorstellung

- Besteht aus
  - einem Sensor
  - drei Boxen
- Box A ist direkt am Sensor angeschlossen
  - stellt Sensorwerte anderen Boxen über den PS-Service *EnergyPS* zur Verfügung
- Box B sammelt Werte und stellt Funktionen über RPC-Service *EnergyRPC* zur Verfügung
  - Durchschnittswert, Minimum und Maximum der letzten x Sensorwerte
- Box C ruft RPCs von Box B über Service-User *EnergyRPCClient* auf

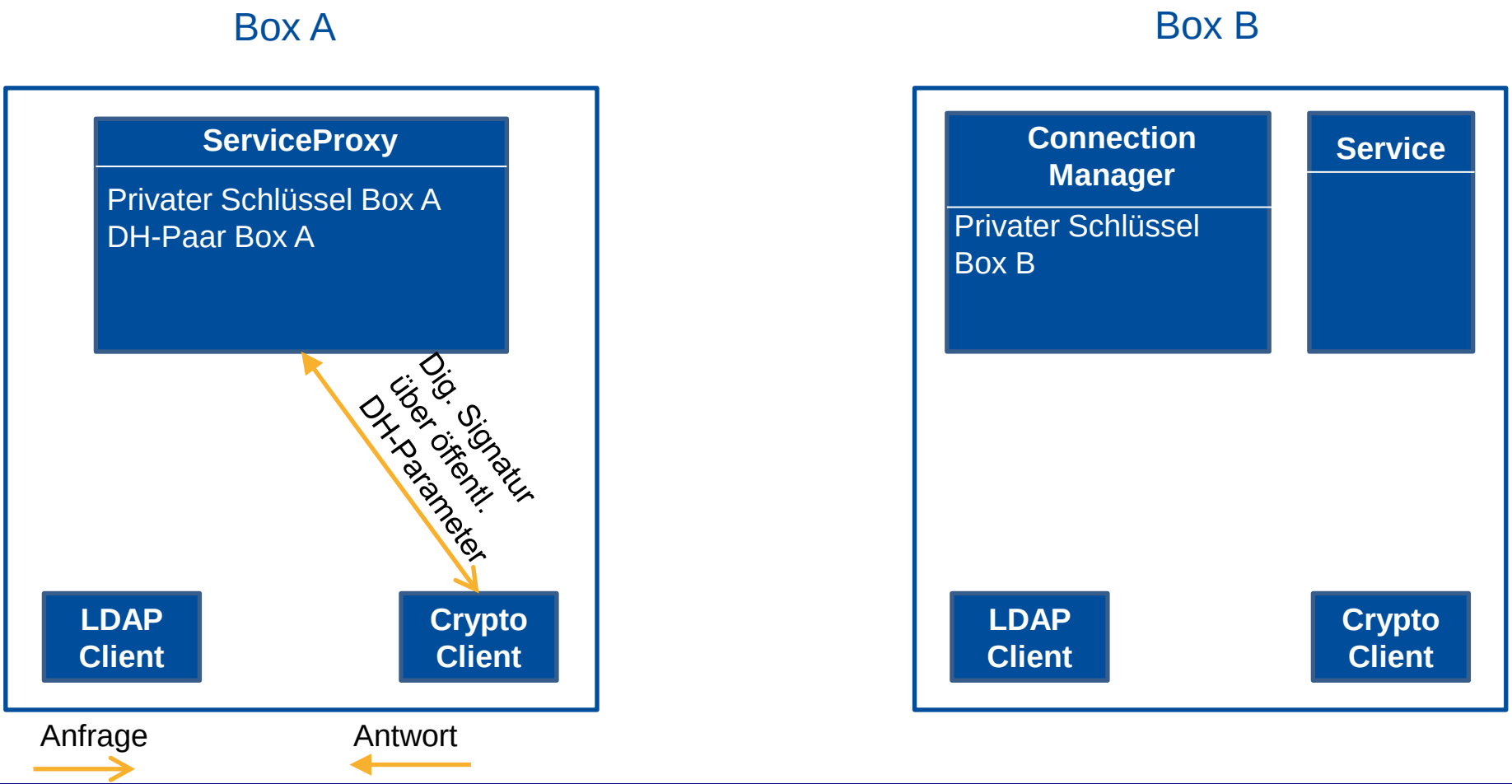
Prototyp Vorstellung



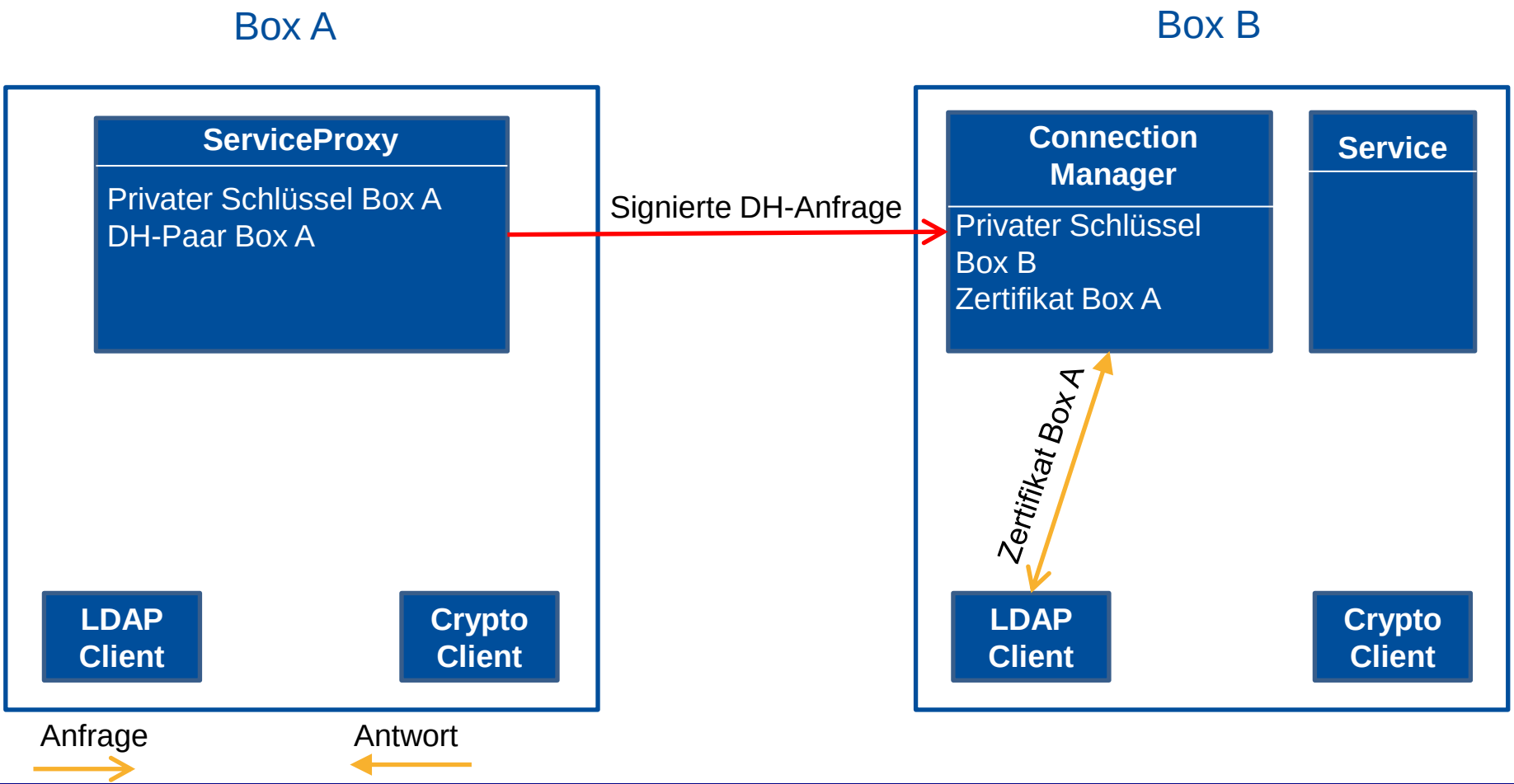
Schlüsselaustausch



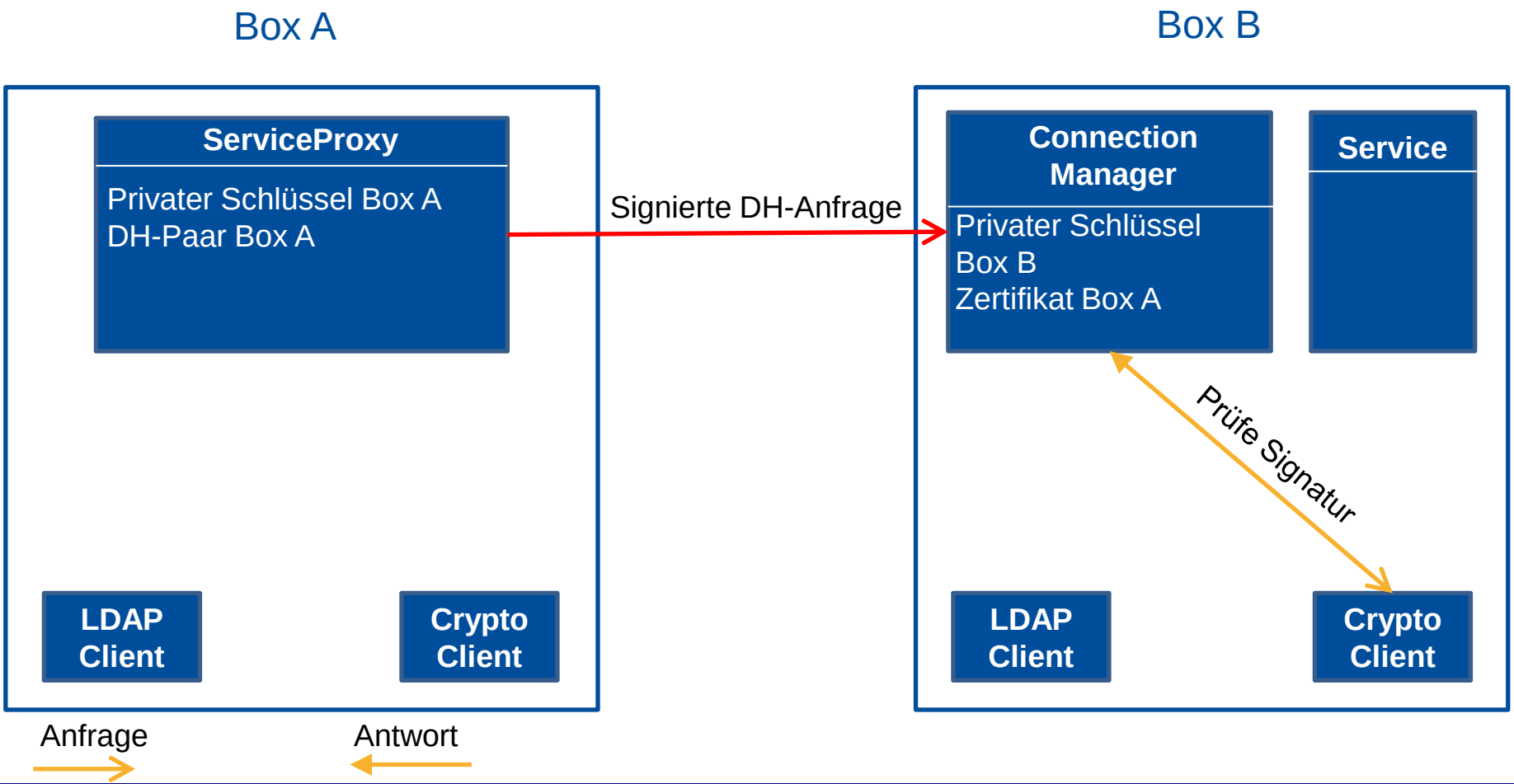
Schlüsselaustausch



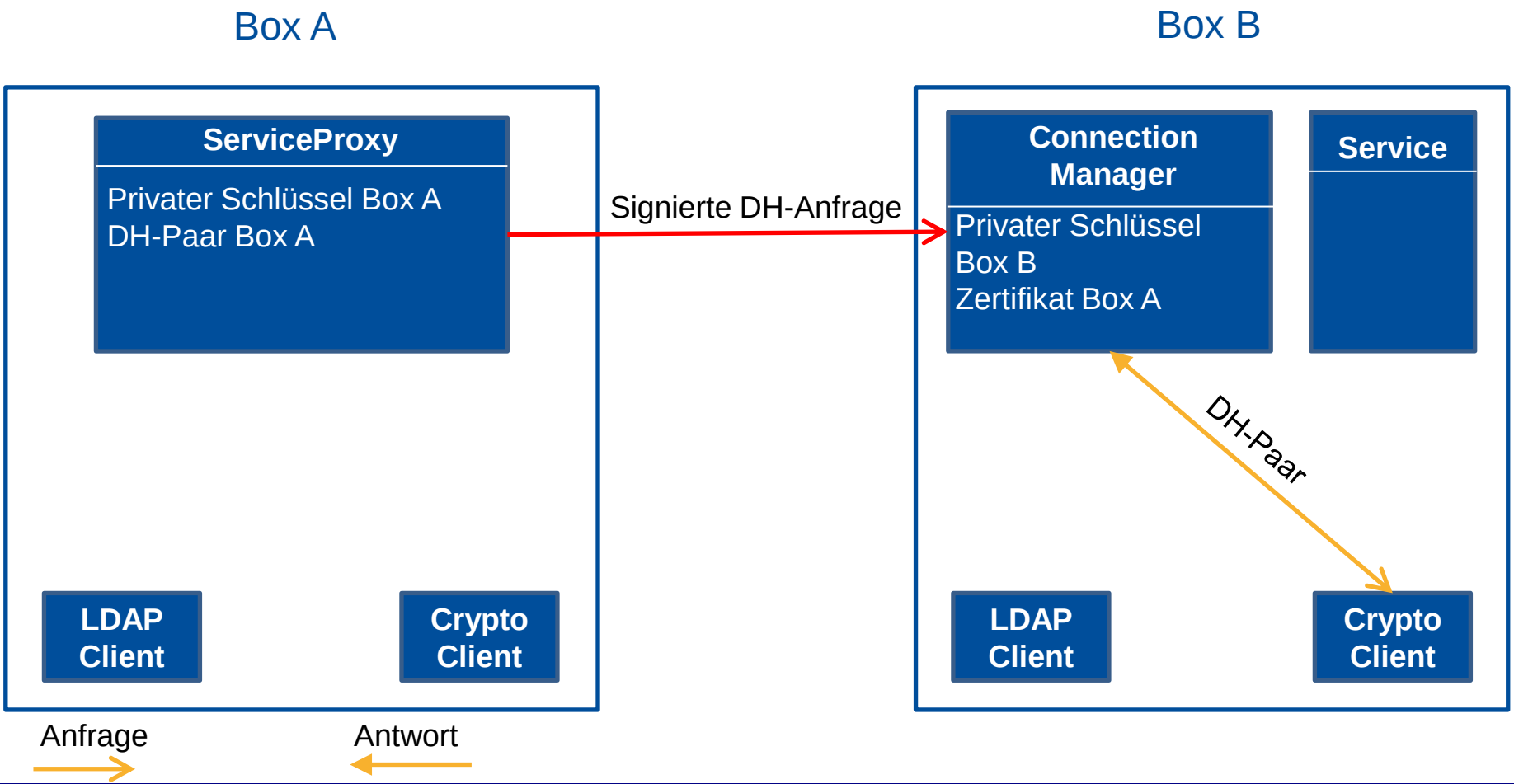
Schlüsselaustausch



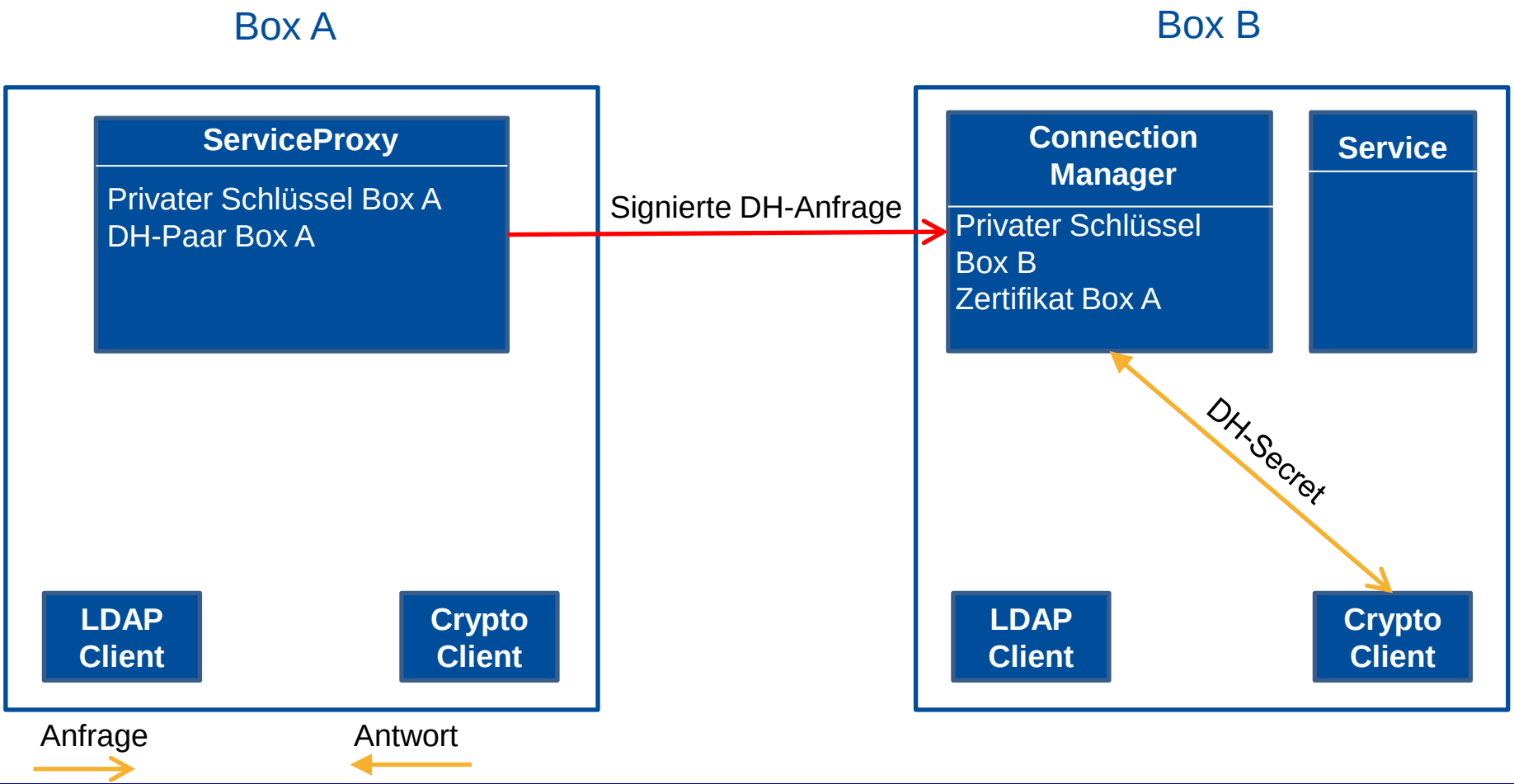
Schlüsselaustausch



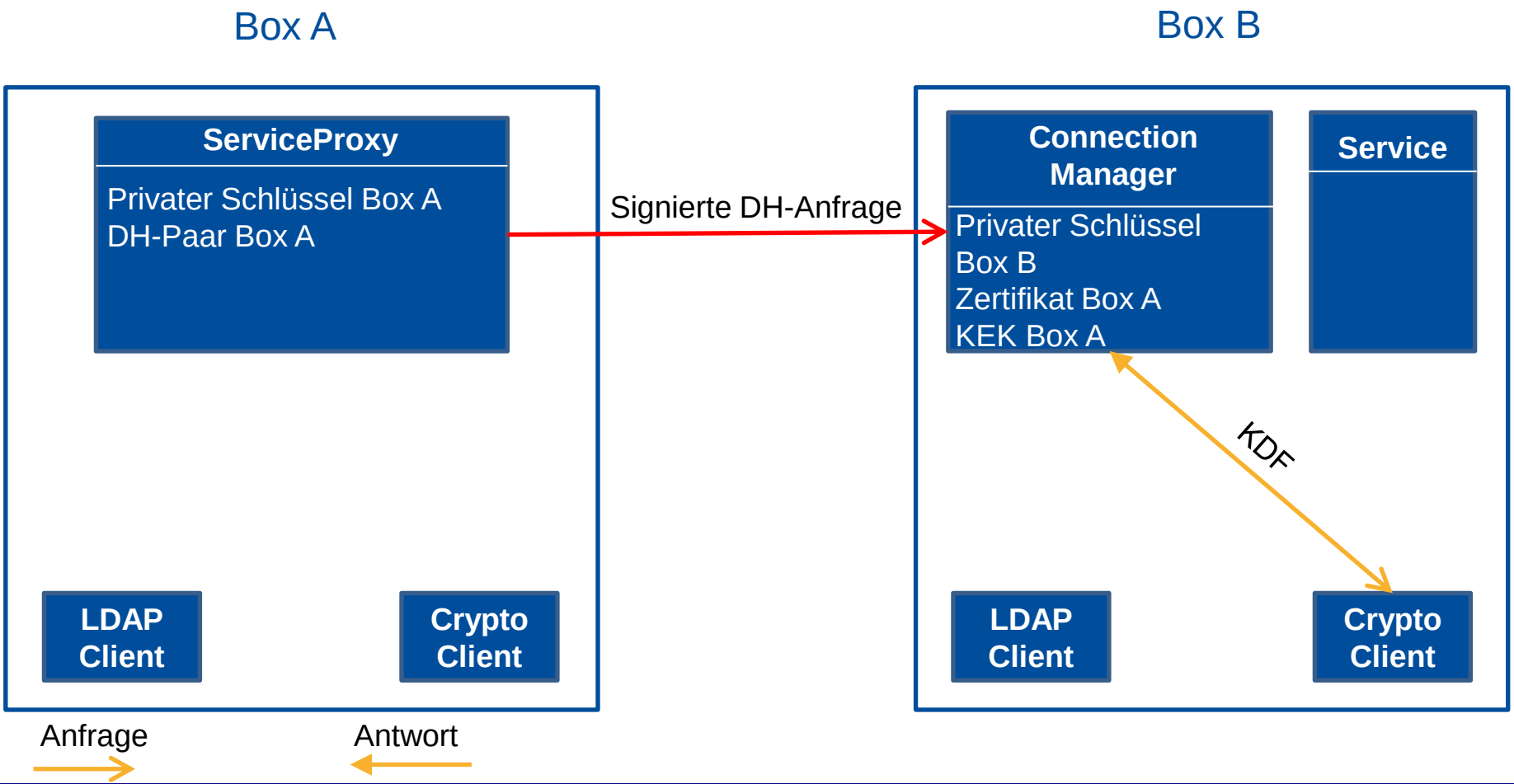
Schlüsselaustausch



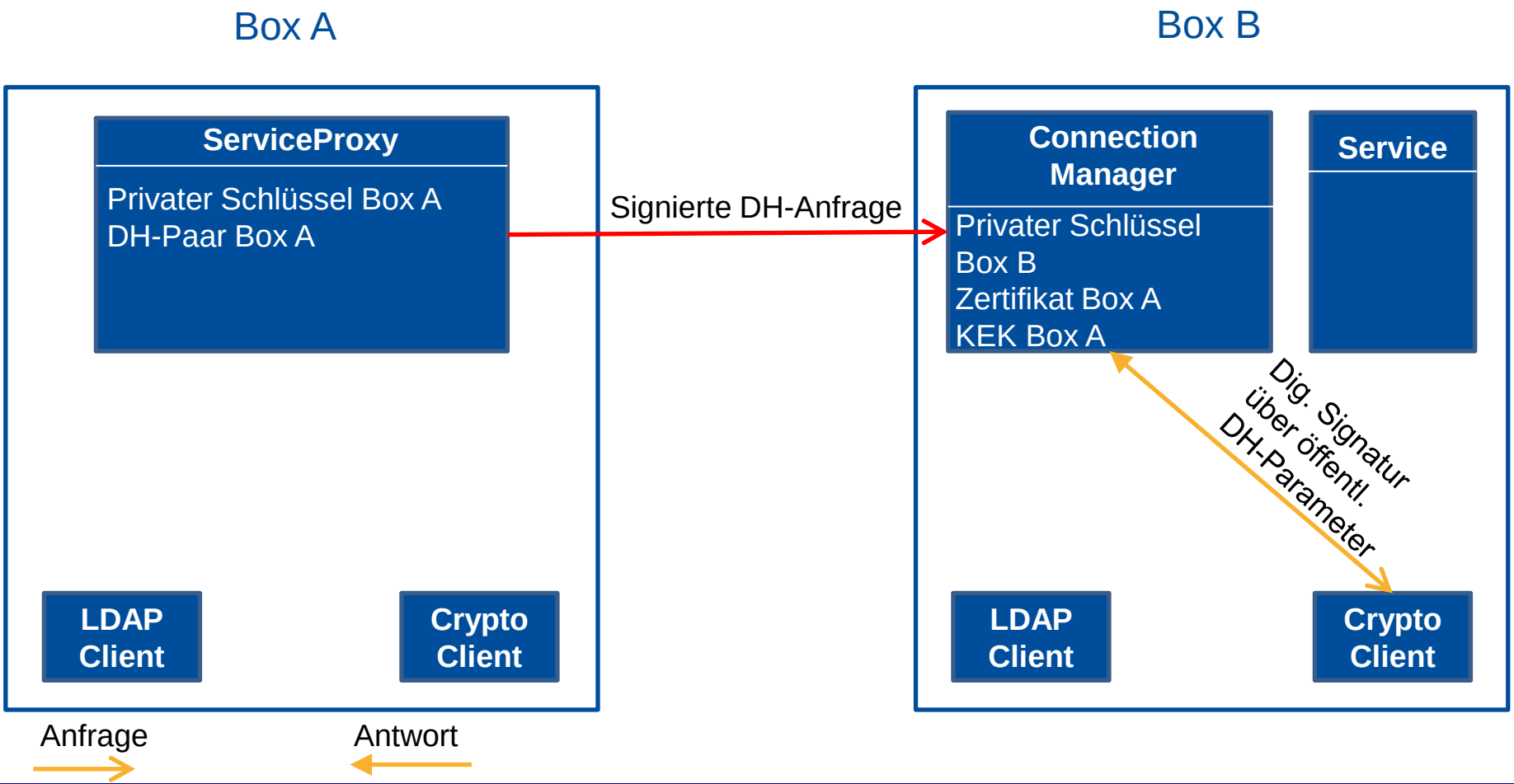
Schlüsselaustausch



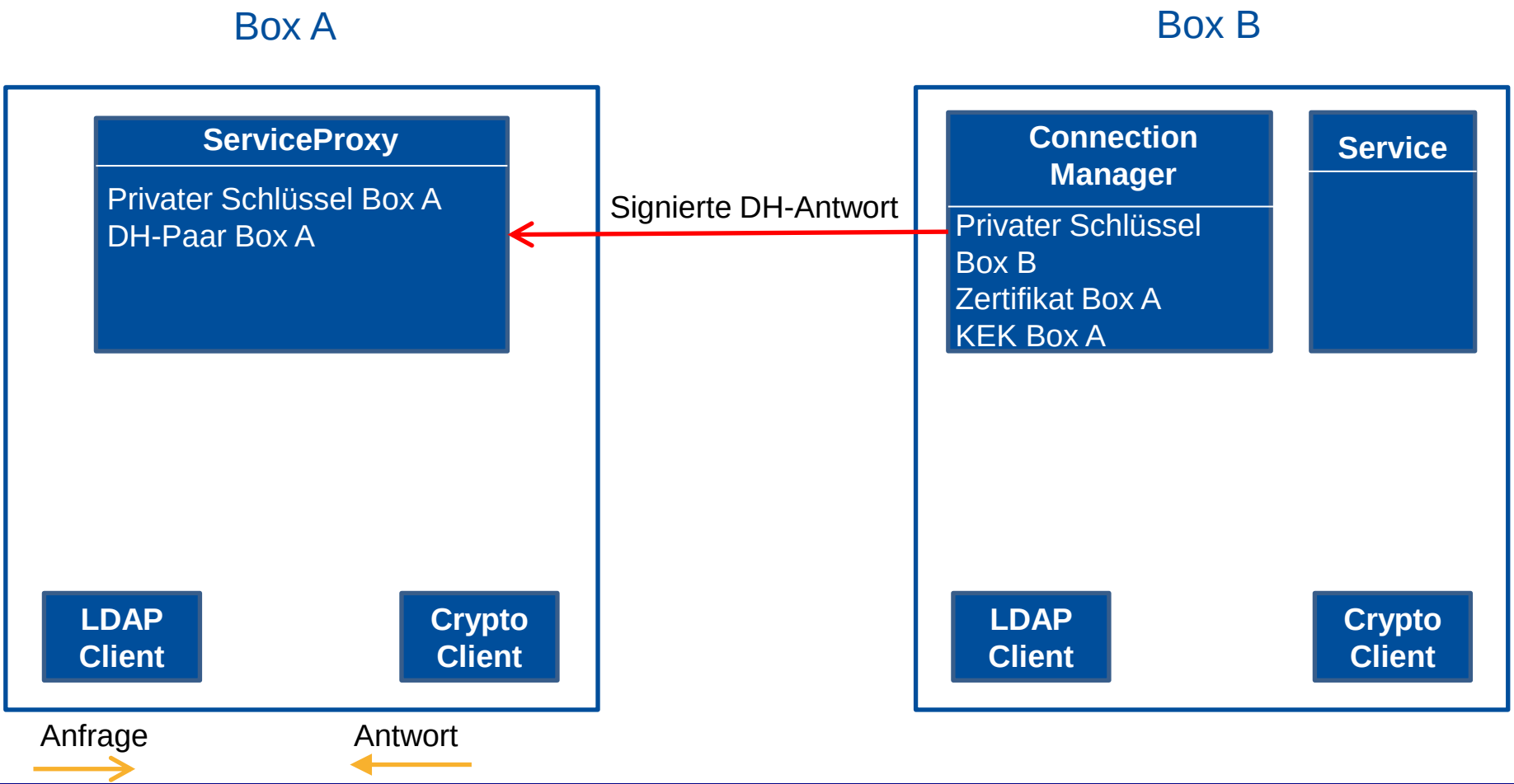
Schlüsselaustausch



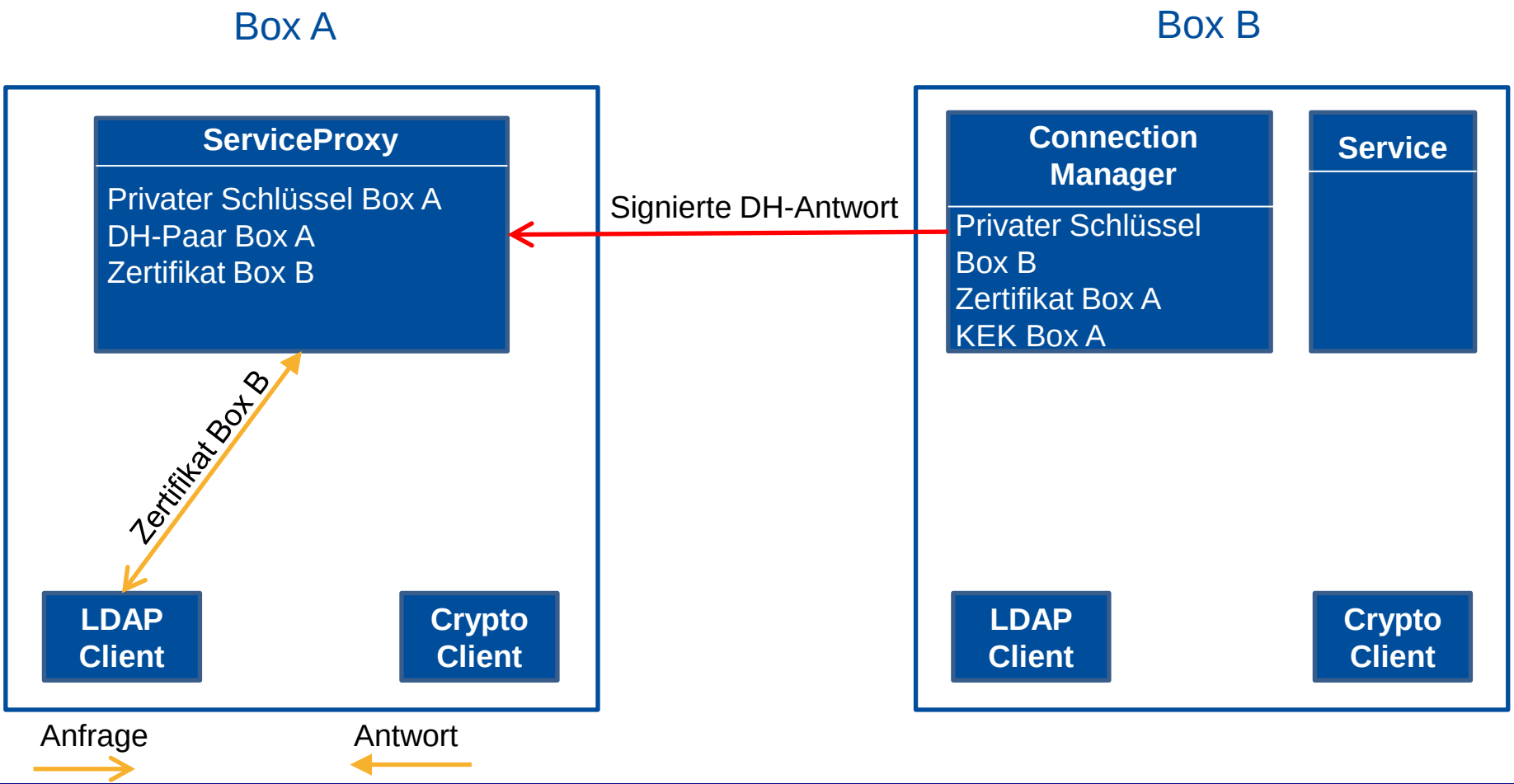
Schlüsselaustausch



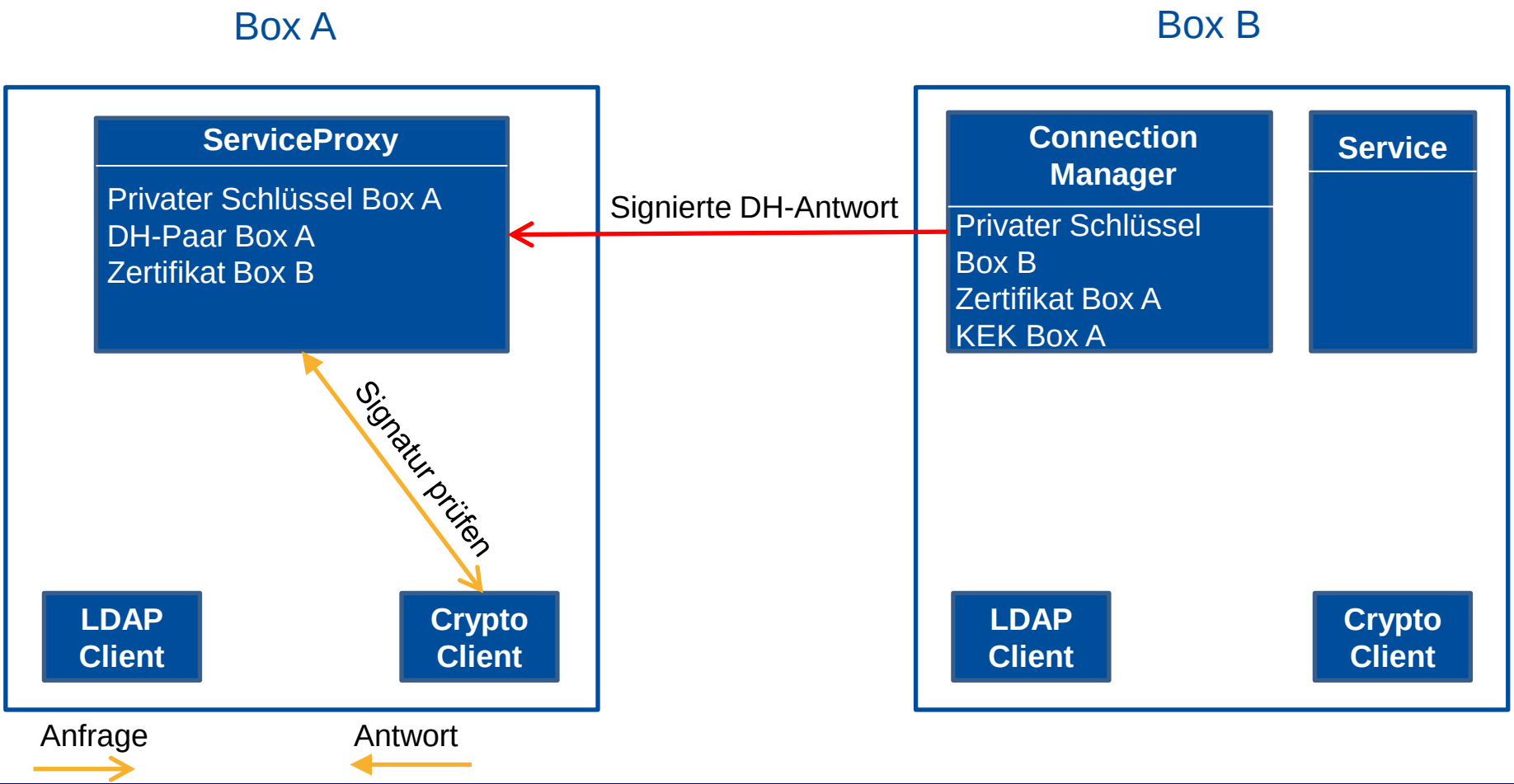
Schlüsselaustausch



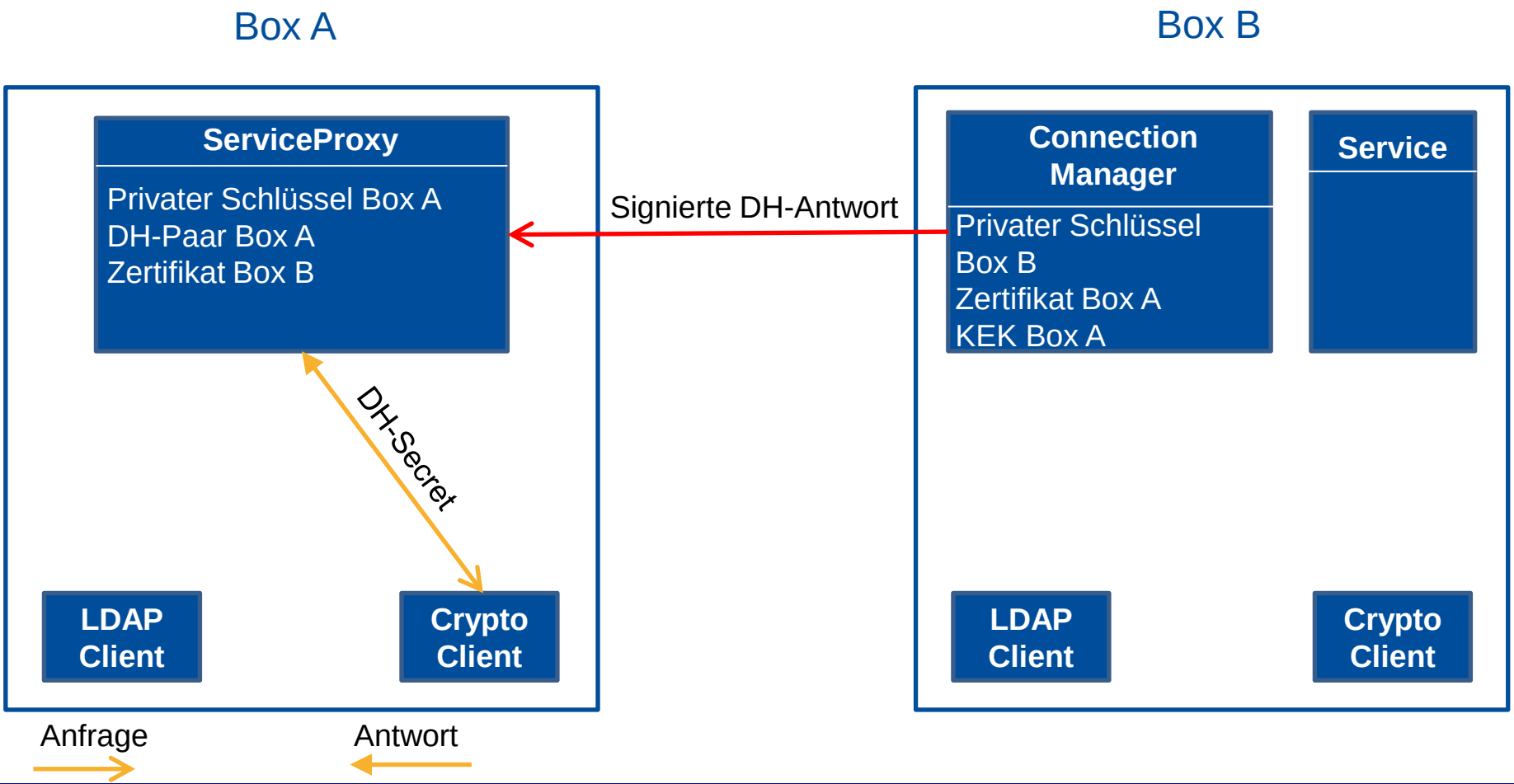
Schlüsselaustausch



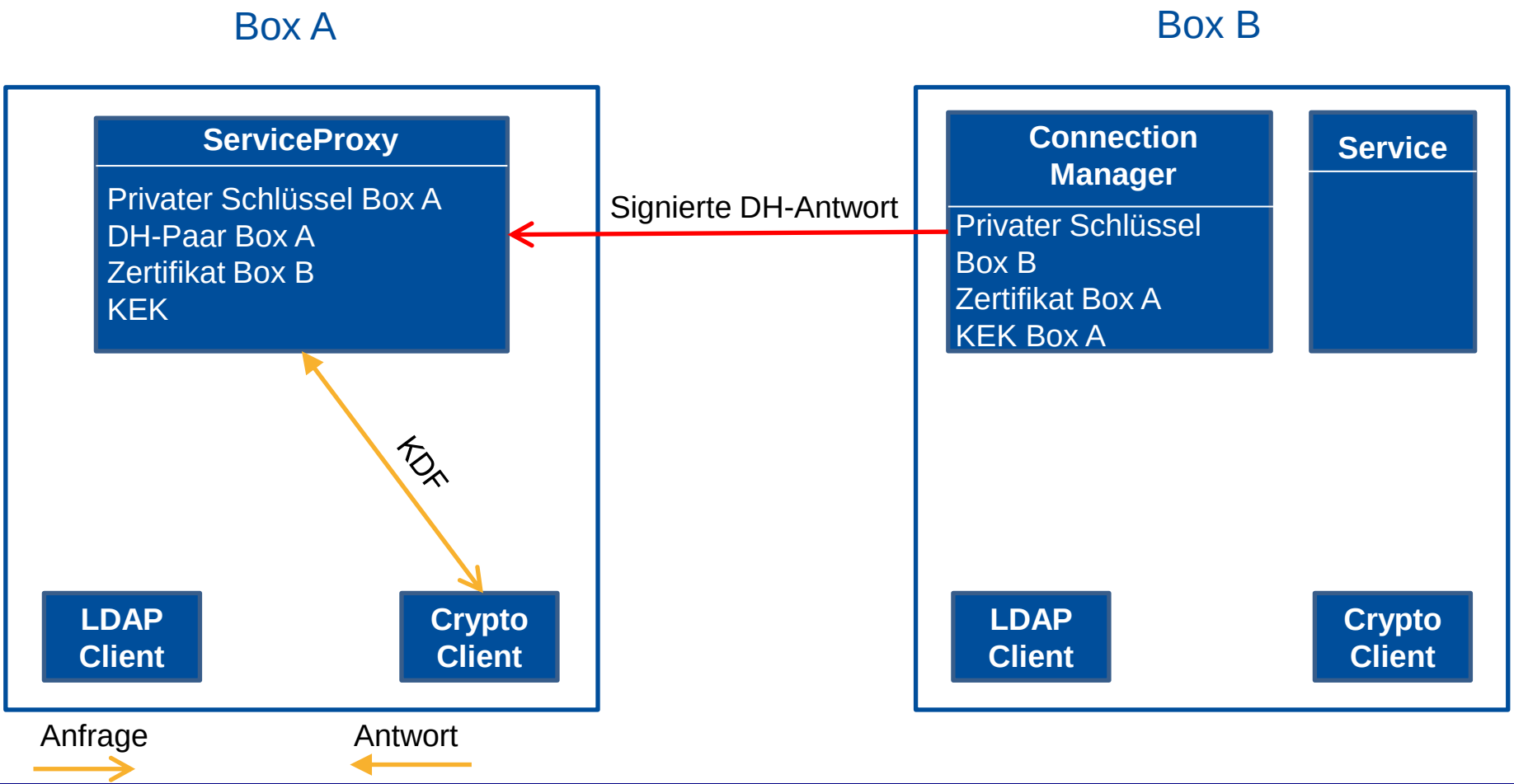
Schlüsselaustausch



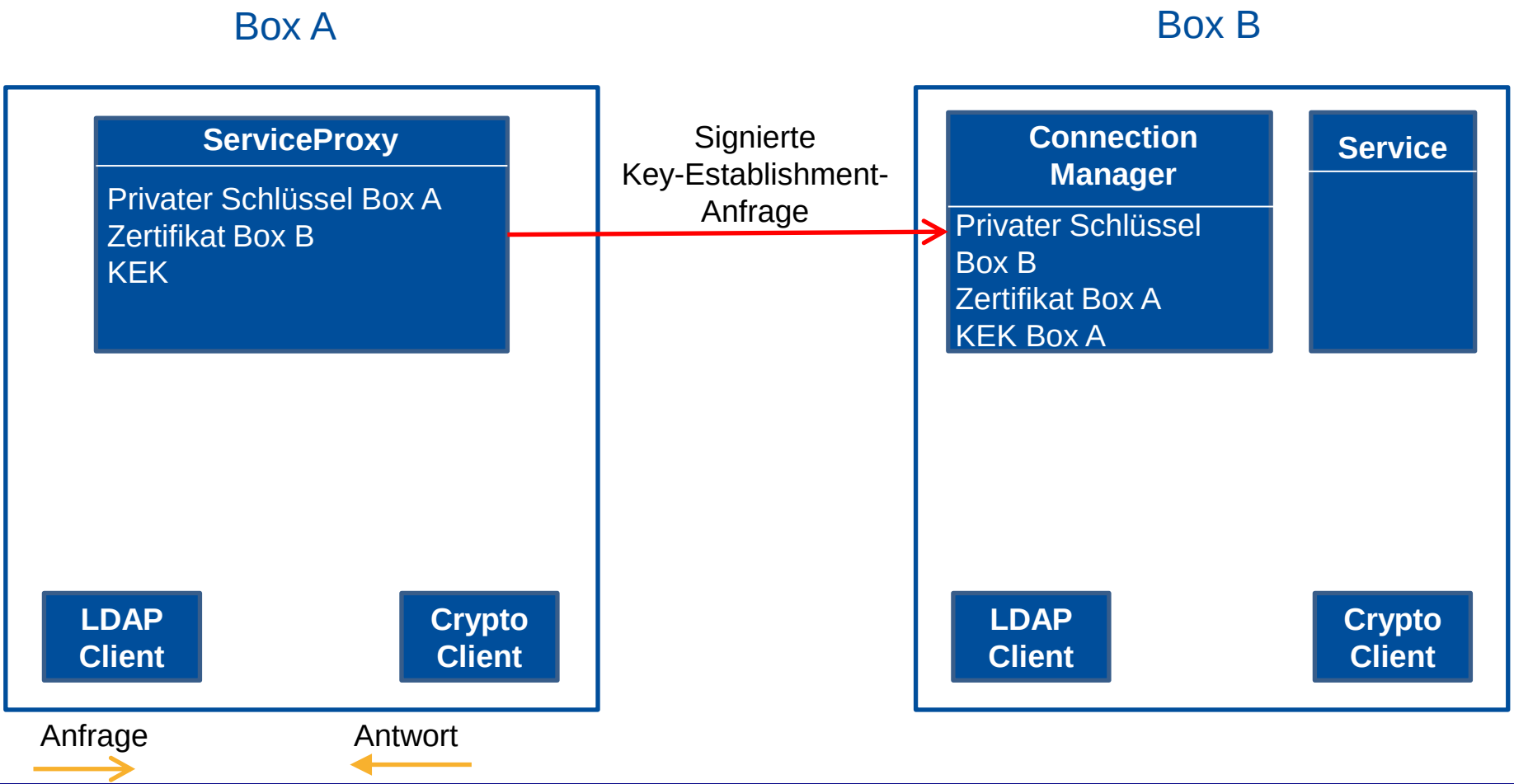
Schlüsselaustausch



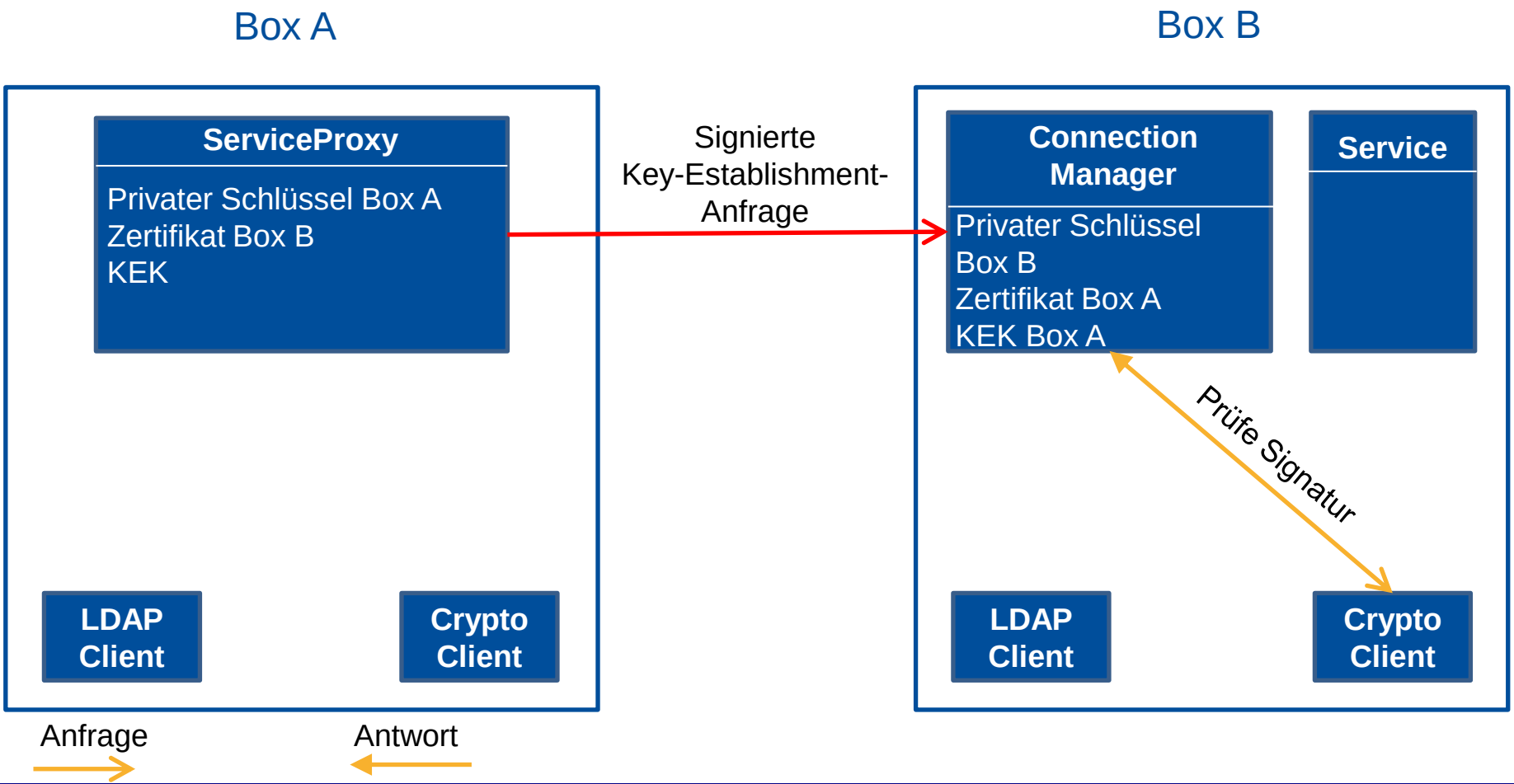
Schlüsselaustausch



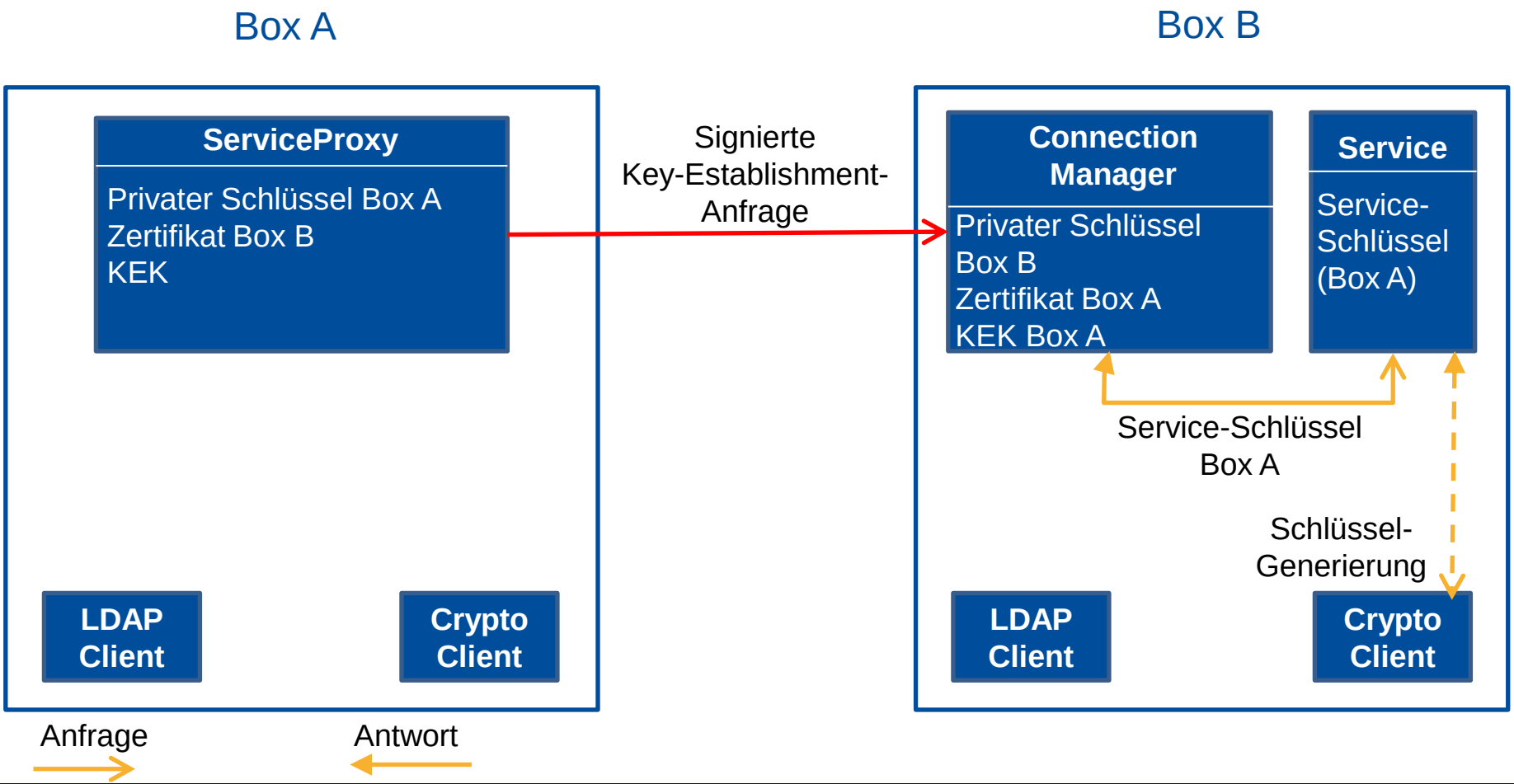
Schlüsselaustausch



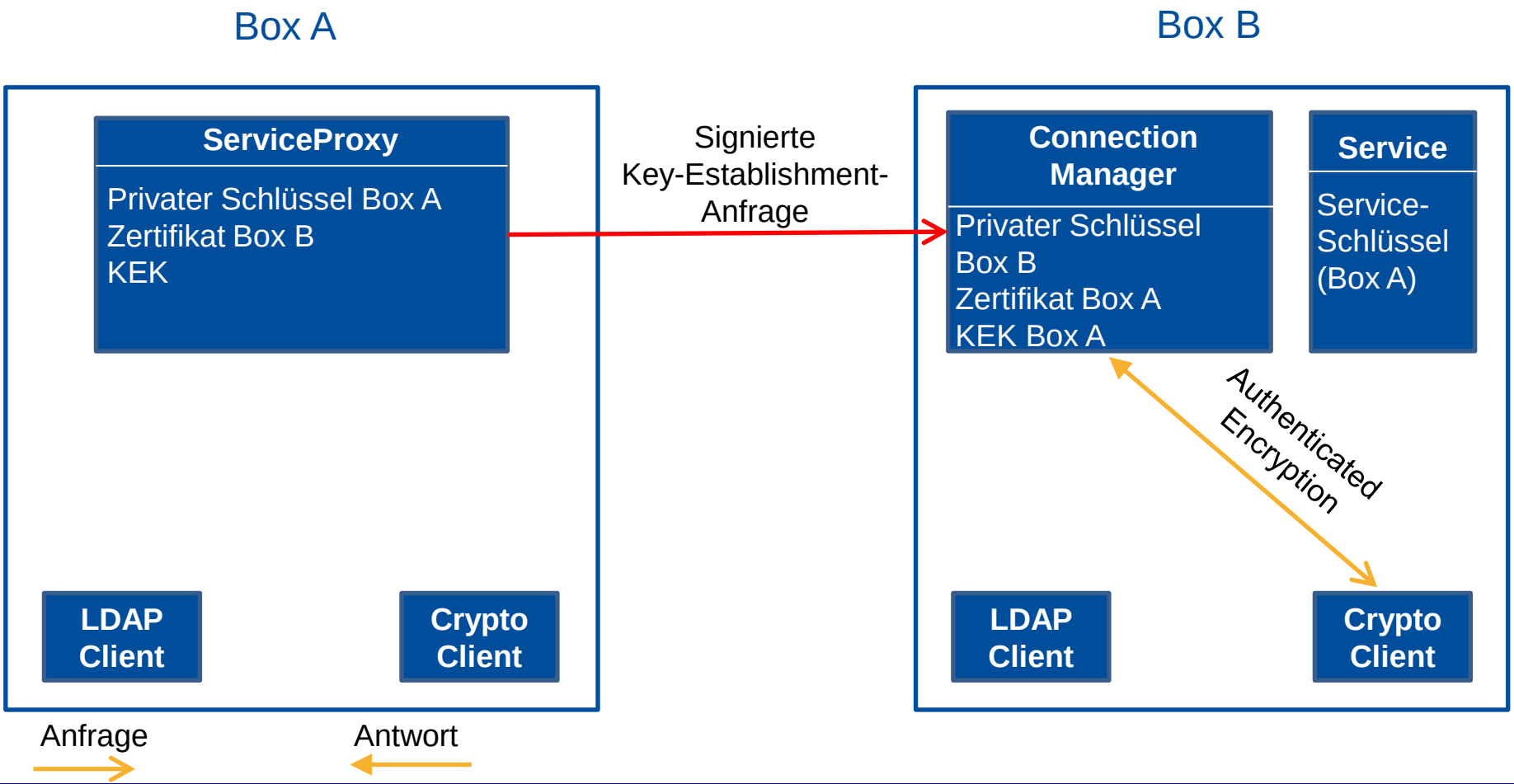
Schlüsselaustausch



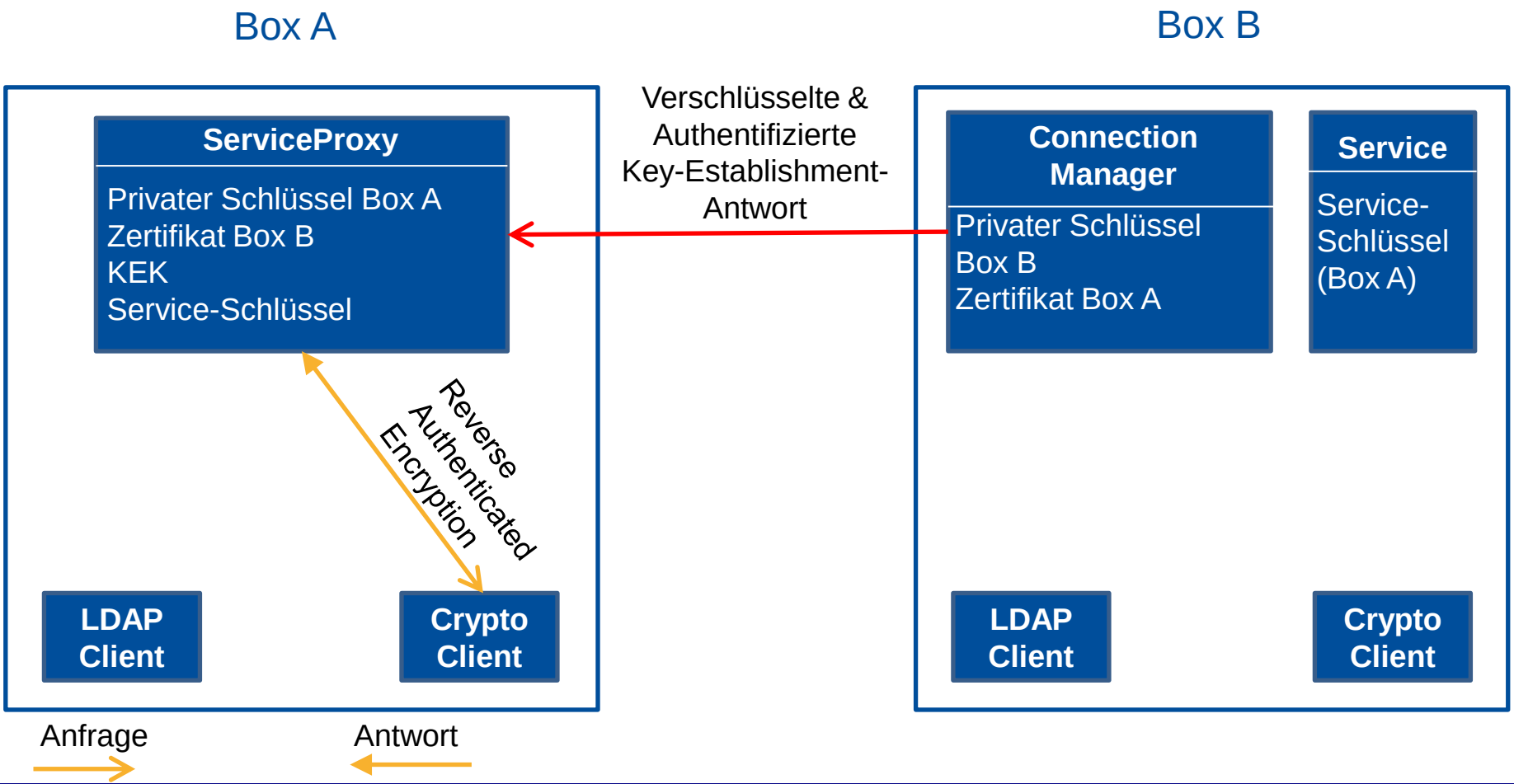
Schlüsselaustausch



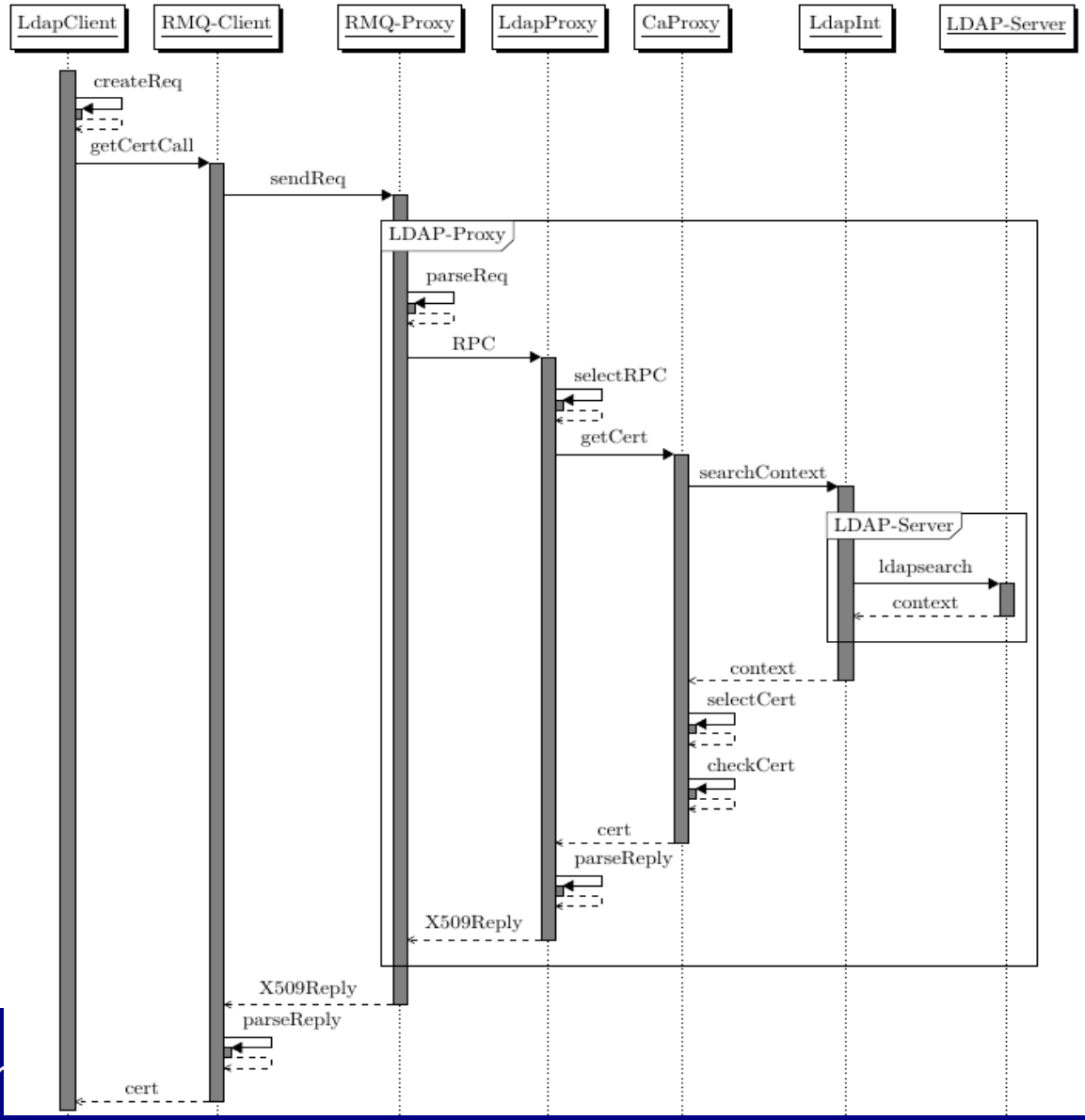
Schlüsselaustausch



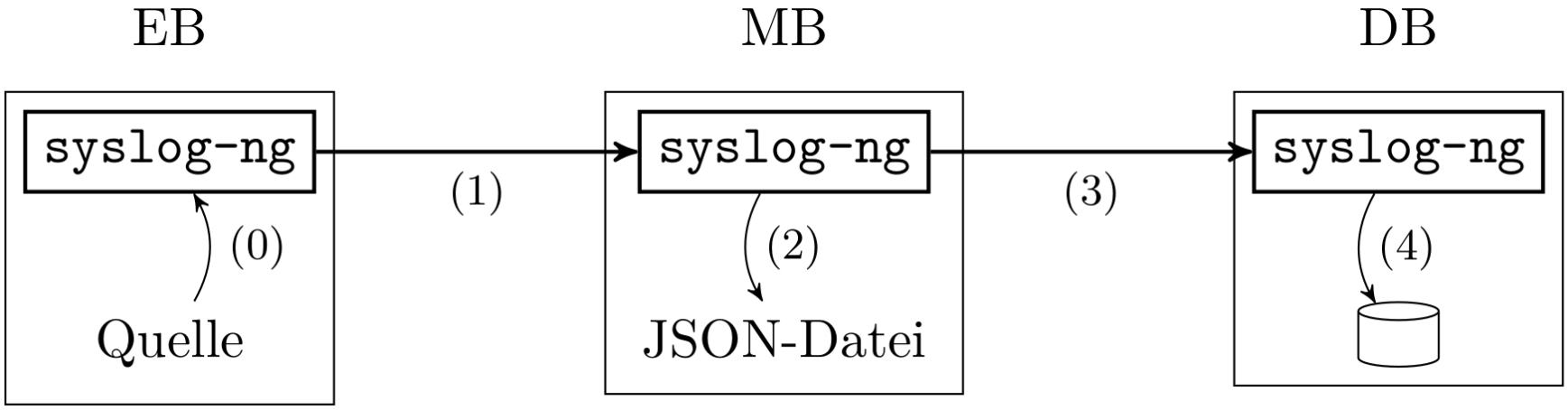
Schlüsselaustausch



LDAP-Proxy: Zertifikat abrufen

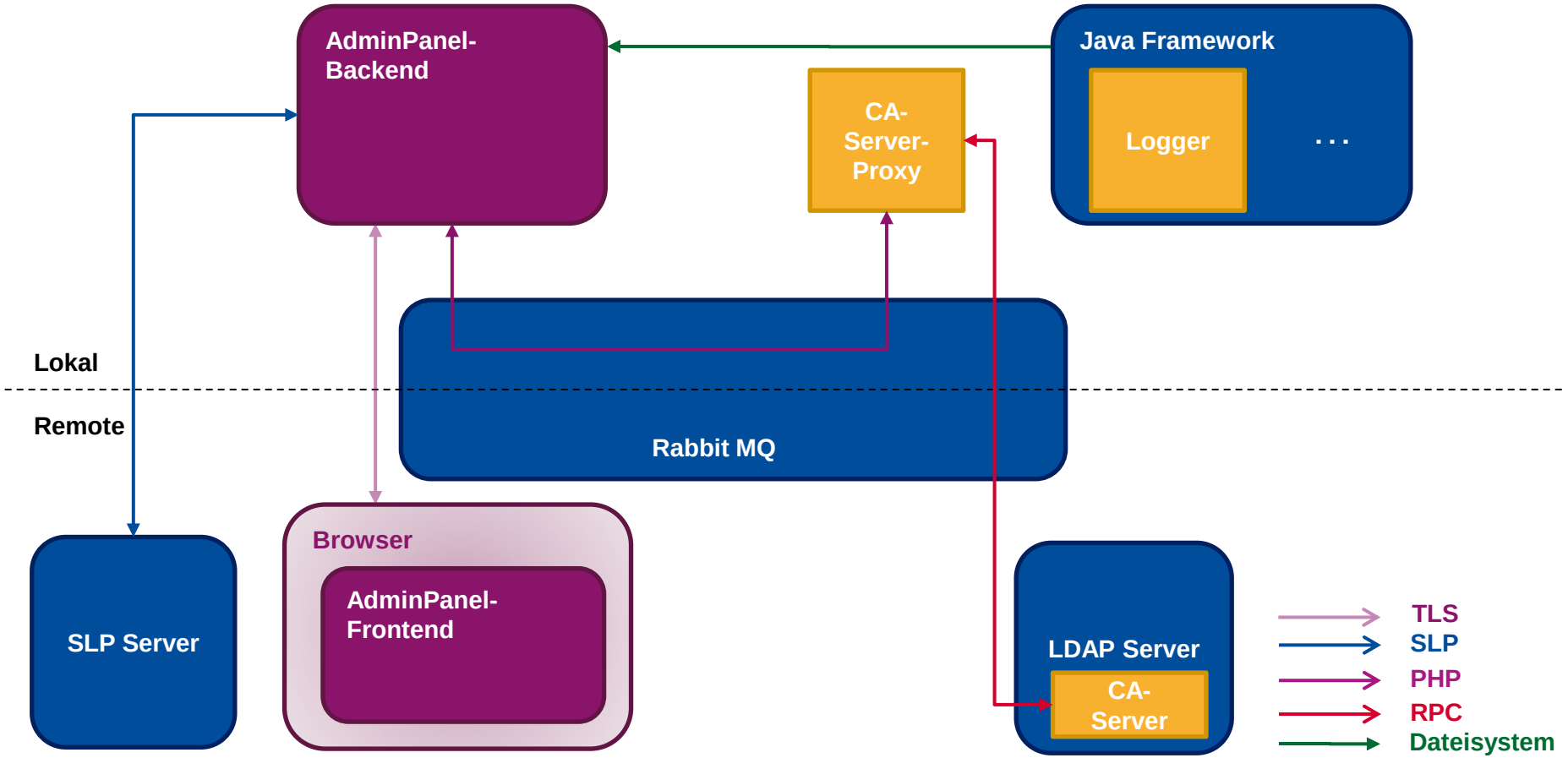


Logging-System



- syslog-ng als zentralisierte Log-Management-Software
- Verbindungen (1) und (3) über TLS gesichert
- JSON-Datei für webbasiertes AdminPanel
- Speichern der Logs in MongoDB

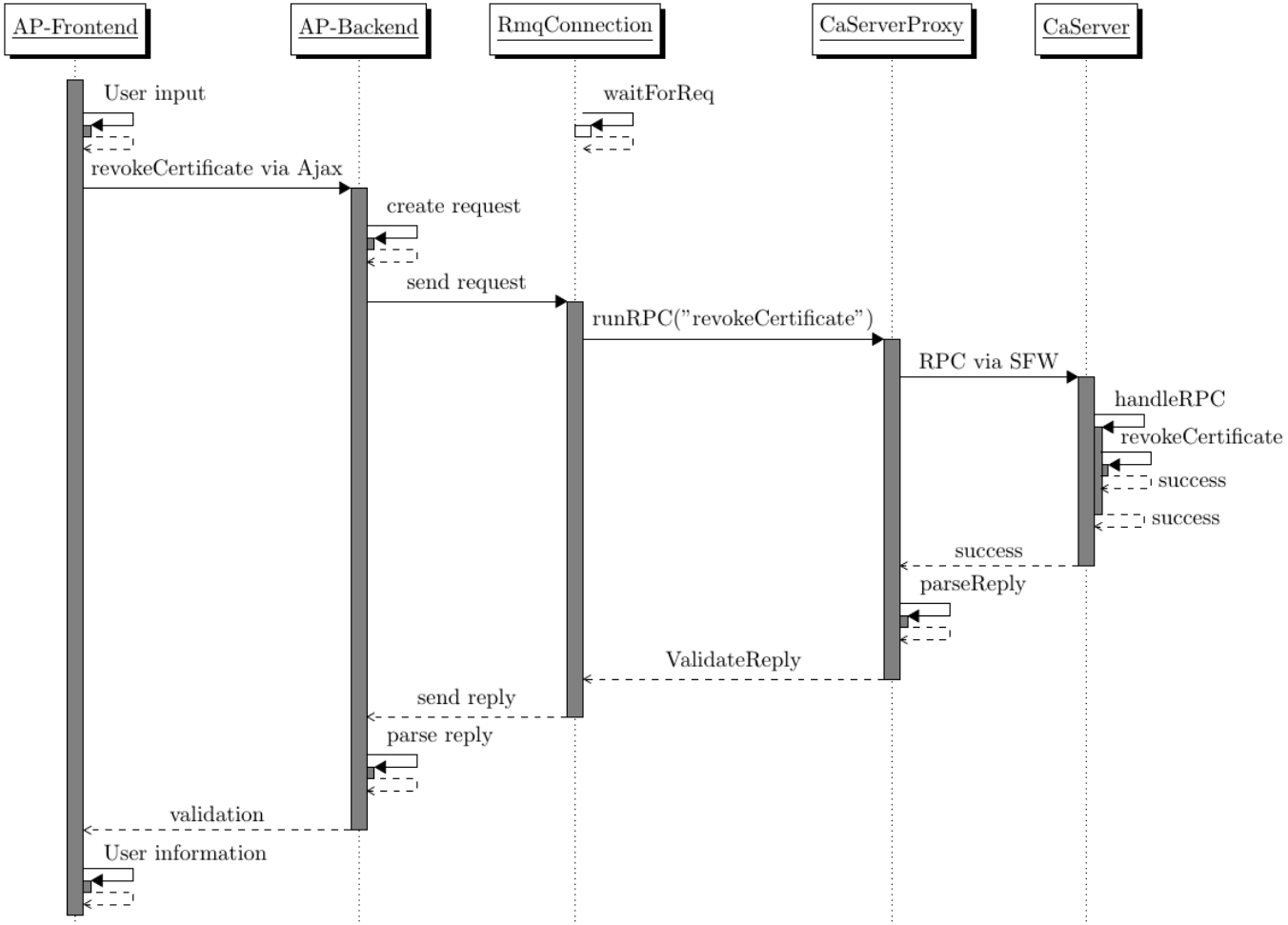
AdminPanel


AdminPanel-Backend
Java Framework
Logger
. . .
CA-Server-Proxy
Lokal
Remote
Rabbit MQ
Browser
AdminPanel-Frontend
SLP Server
LDAP Server
CA-Server
TLS
SLP
PHP
RPC
Dateisystem

Funktionen des AdminPanel

- Statusanzeige
- Konfiguration
- Logs anzeigen
- Zertifikat widerrufen
- Zertifikat aktualisieren

Zertifikatswiderruf im Framework



Zusammenfassung

- Definition einer Netzwerktopologie
- Entwicklung einer Softwarearchitektur
- Erstellung eines Sicherheitskonzepts
- Prototypische Implementierung der Softwarearchitektur und des Sicherheitskonzepts

Live Demo

- Video
  - Zeigt die notwendigen Schritte um eine Anwendung
    - Zu entwickeln
    - Auf einer Box einzurichten
  - Zeigt die Ausführung des Service-Frameworks