

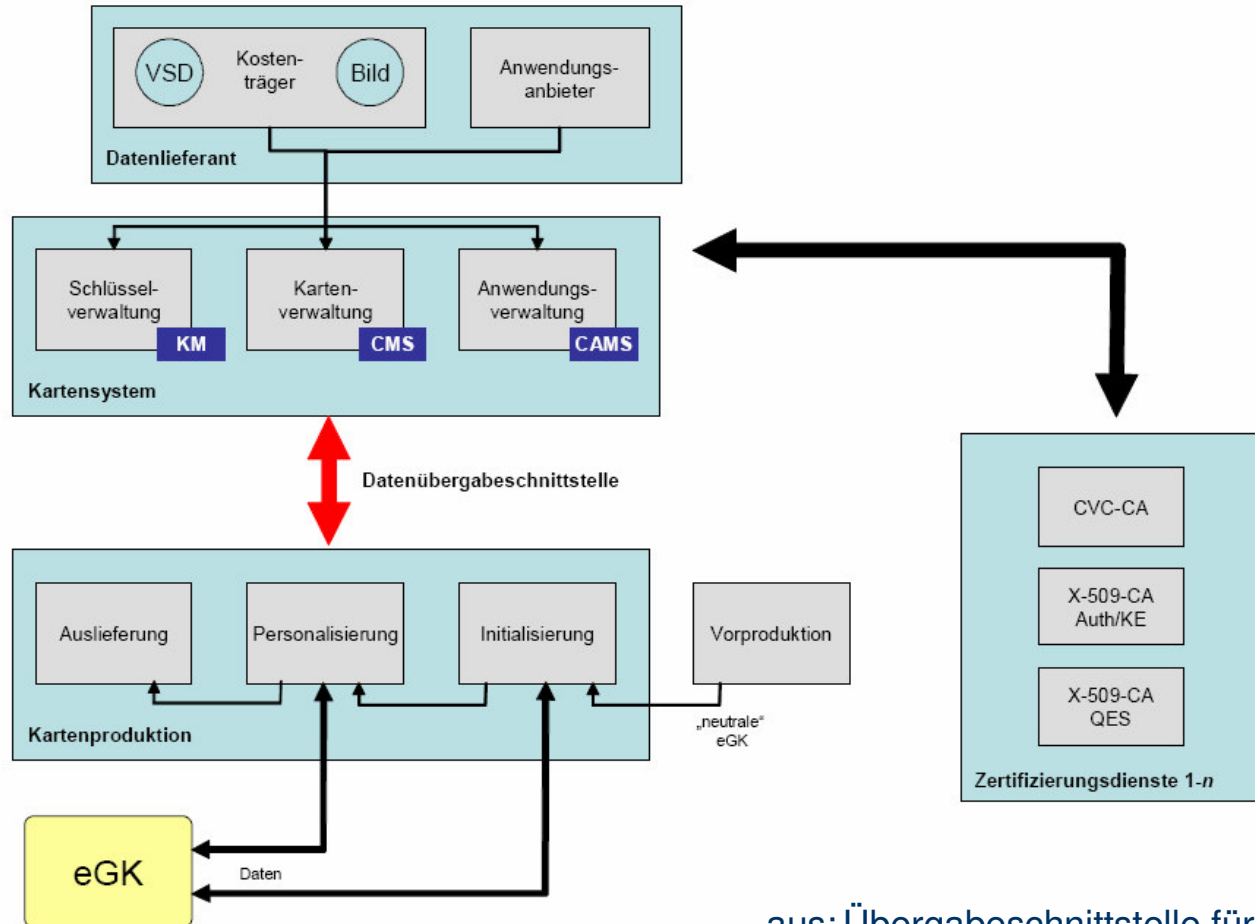
Design moderner, wirtschaftlicher PKI-Plattformen im Kontext der Anforderungen aktueller Großprojekte

Erik Neumann
eneumann@mtg.de
media transfer AG, Darmstadt

Inhalt

- Aufbau der PKI der eGK
 - Komponenten/Systeme
 - Zertifikatshierarchie
- Anforderungen
 - Wiederverwendbarkeit / Portabilität
 - Stückzahlen
 - Synchrones Zertifikatsmanagement
 - Zertifikatsverwaltung
 - Externe Schnittstellen
 - Modularität
 - Skalierbarkeit
 - Administration
- Architekturvorschlag
 - Komponenten und Module
 - Skalierbarkeit

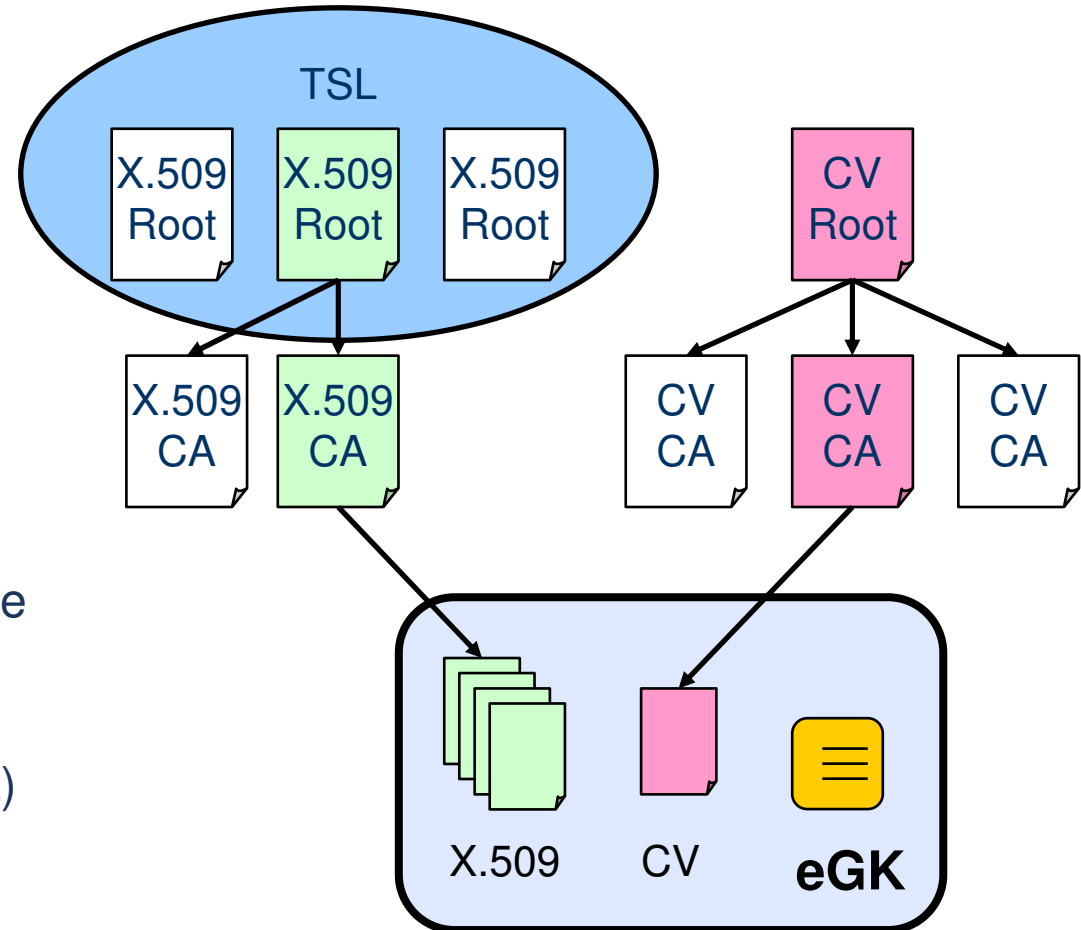
Aufbau der PKI der eGK: Systeme/Komponenten



aus: Übergabeschnittstelle für die Produktion der eGK, Version 1.1.1, 07.09.2006, gematik

Aufbau der PKI der eGK: Zertifikatshierarchie

- vereinfachte Darstellung (ohne QES-Zertifikate, keine HBAs, nur eine X.509 CA je Karte)
- vier X.509-Zertifikate und ein CV-Zertifikat je eGK
- zweistufige Hierarchien
- gematik betreibt CV-Root
- eine CV-CA je Betreiber
- eigene X.509-Hierarchie je Betreiber, d.h. eine Root und evtl. mehrere CAs
- Trusted Service List (TSL) der zugelassenen X.509-Roots durch die gematik



Anforderungen: Wiederverwendbarkeit/Portabilität

- hohe einmalige Investition (insb. wegen Spitzenlast bei Erstausrüstung)
 - Nutzung der gleichen Software für andere PKIs
z.B. interne PKI des Betreibers
 - Flexible Verfahren zur Zertifikatsbeantragung und –erzeugung
PKCS#10, SPKAC, unsignierte Schlüssel, Schlüsselgenerierung usw.
 - Flexible Zertifikatsformate und -medien
Softwarezertifikate, Chipkarten, USB-Tokens
- hohe laufende Unterhaltskosten
 - Unabhängigkeit von Architekturen oder gar Produkten
 - Verwendung von verbreiteten Hochsprachen
 - standardisierte und verbreitete Schnittstellen/Protokolle
 - Kapselung produkt-spezifischer Schnittstellen (z.B. DB, HSM)
 - Verwendung der gleichen Infrastruktur für weitere PKIs
z.B. gemeinsame Nutzung durch mehrere Krankenkassen

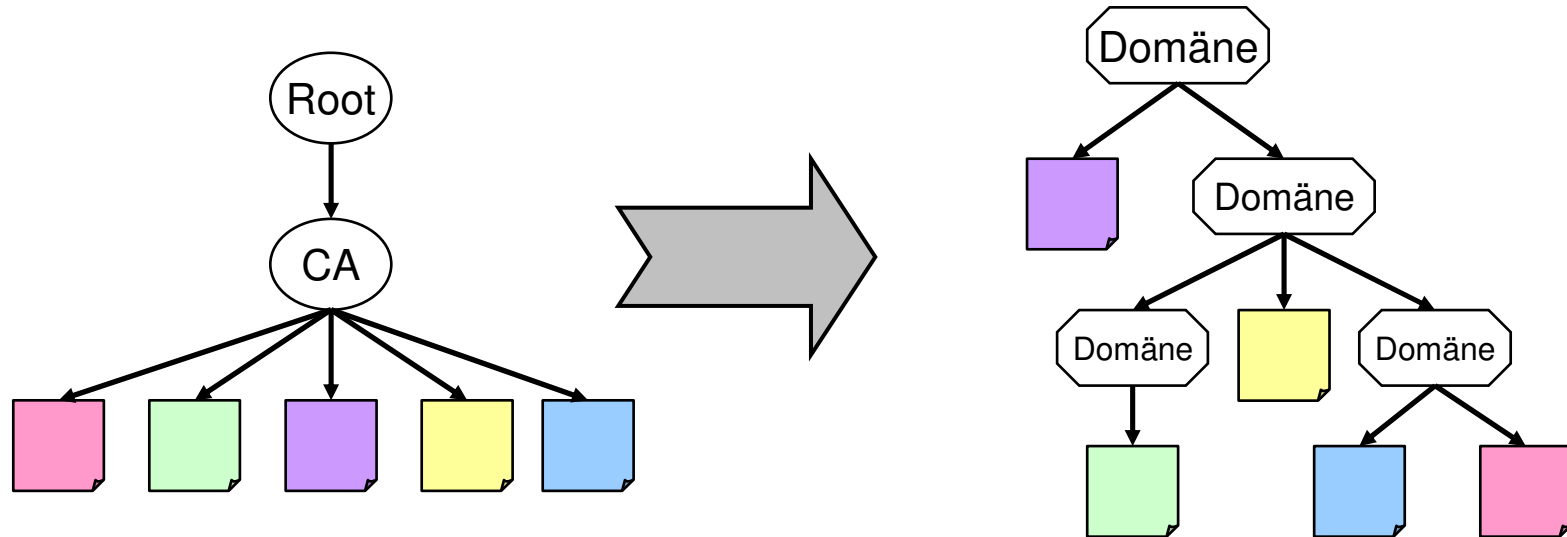
Anforderungen: Stückzahlen

- „Quantität als eigene Qualität“
- Kritisch sind Kartenproduktion (physischer Prozess) und kryptografische Algorithmen (rechenintensiv)
- Krankenkasse mit 10 Millionen Versicherten (40h/Woche vs. 24x7)
 - Erstaussstattung innerhalb von 6 Monaten
 - 38-174 Karten bzw. 190-868 Zertifikate pro Minute
 - Dauerbetrieb (Laufzeit 4 Jahre, Fluktuation von 5% pro Jahr)
 - 5,7-26 Karten bzw. 28,5-130 Zertifikate pro Minute
 - OCSP-Anfragen (16 Arztbesuche pro Patient und Jahr)
 - 1.388 Anfragen pro Minute (bei 40h/Woche)
- Parallelisierung bedingt Vervielfältigung der CA-Schlüssel
- Zusätzliche Prozesse (z.B. Abrechnung, Export) auf anderer Infrastruktur
- Eignung User-Interface, keine manuellen Prozesse

Anforderungen: Synchrones Zertifikatsmanagement

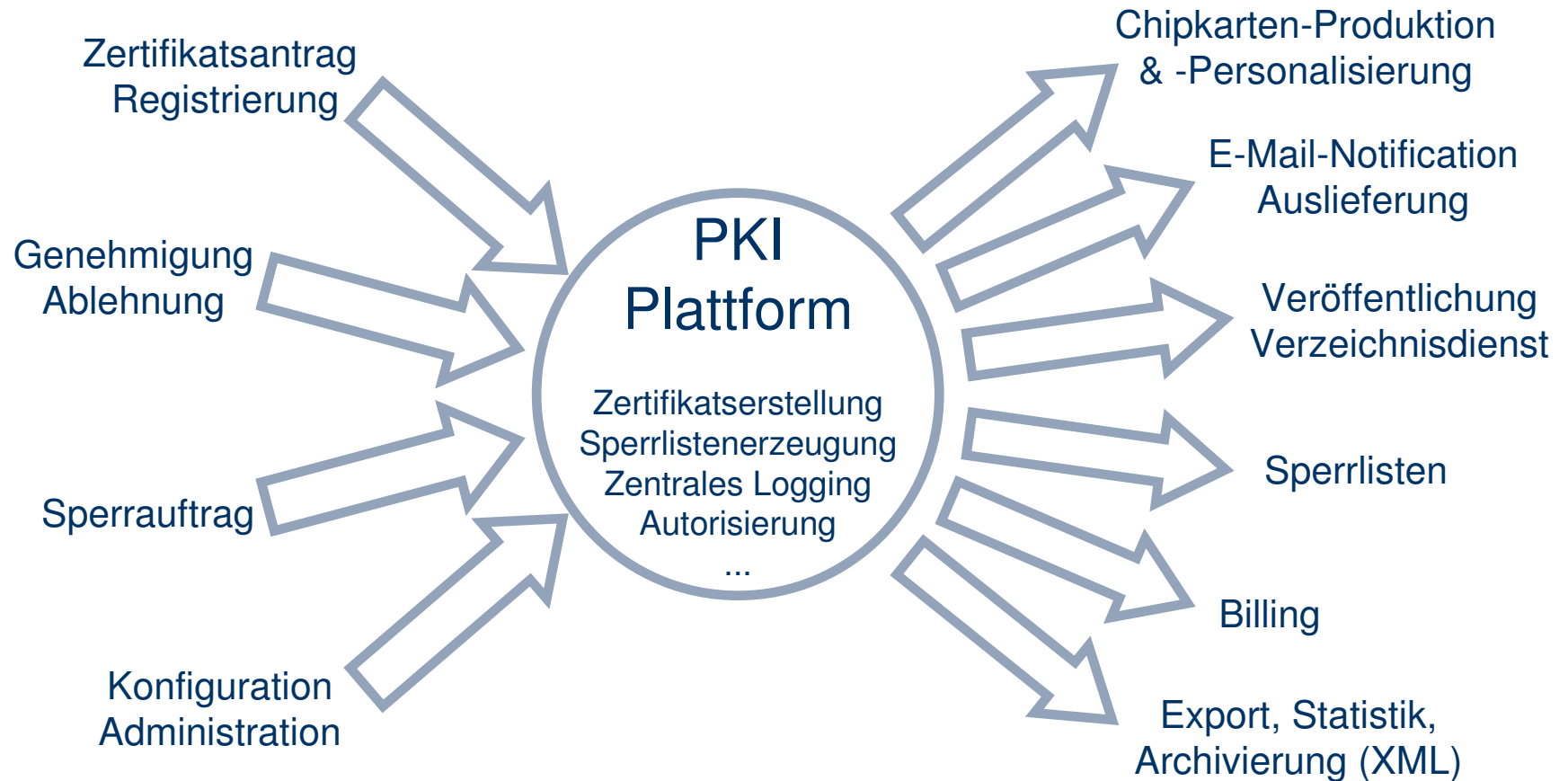
- Eine eGK beherbergt ein CV-Zertifikat, vier fortgeschrittene X.509-Zertifikate, und optional ein qualifiziertes X.509-Zertifikat.
- Die Zertifikatsverwaltung unterscheidet nicht die Zertifikate, sondern die Karten, d.h. die Plattform sollte die Zertifikate einer Karte immer „gemeinsam“ (synchron) bearbeiten.
 - Abbildung „realer“ Zertifikate auf „logische“ Zertifikate (=Chipkarten)
 - unterschiedliche Zertifikatsformate (X.509 und CV) auf einer Karte
 - Zertifikate von unterschiedlichen CAs auf einer Karte
 - Im Regelfall nur gemeinsame Bearbeitung möglich
z.B. Ausstellung oder Sperrung
 - Aber: Getrennte Bearbeitung in Einzelfällen, z.B. Veröffentlichung
 - Schnittstellen zu externen CAs (QES) zur Synchronisierung

Anforderungen: Zertifikatsverwaltung



- Zertifikatshierarchie nur zweistufig (bei CV-Zertifikaten effektiv einstufig)
- Zertifikatsinhalte definiert und als Gliederungskriterien unzureichend
- Domänen als Verwaltungsstruktur und Grundlage der Rechteverwaltung z.B. Filiale, Wohnort, Sachbearbeiter, Versicherungsschutz o.ä.
- Domäne enthält Zertifikate verschiedener Typen, Formate und CAs
- ermöglicht Mandantenfähigkeit

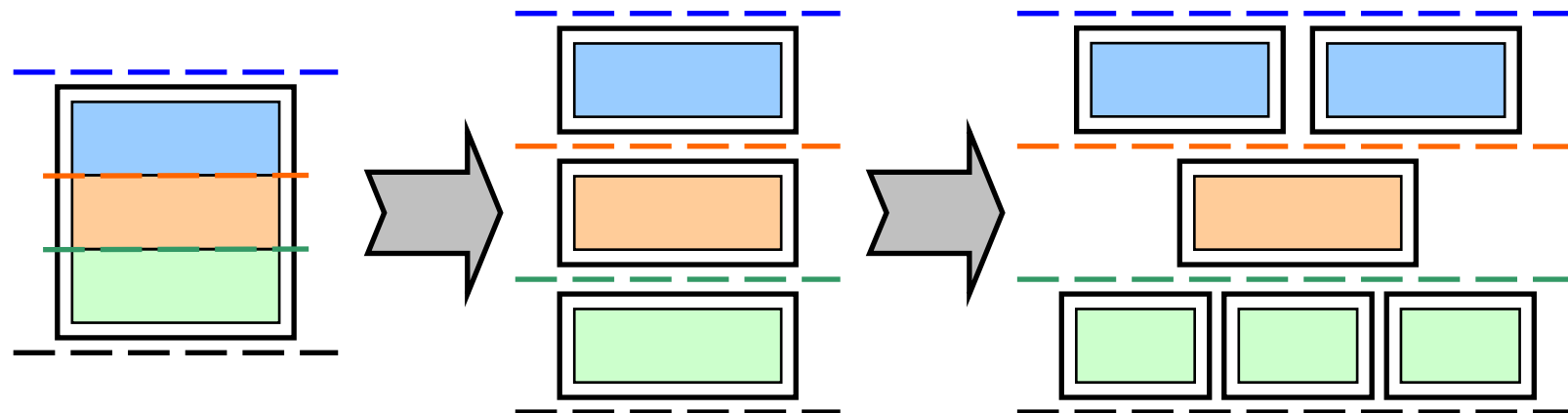
Anforderungen: Externe Schnittstellen



Anforderungen: Modularität

- Einsatz für andere PKIs mit geringeren Anforderungen
 - Auslagerung von Funktionalität in Module, die bei Bedarf weggelassen werden können (geringere Systemanforderungen)
 - E-Mail-Benachrichtigungen
 - LDAP-Server und OCSP-Responder
 - SCEP, CMP usw.
 - Produkte austauschbar durch Abstraktion der Schnittstellen
 - Datenbank: Cluster, Open-Source, Datei-basiert
 - Signaturerstellungseinheit: Software, Chipkarte, Netzwerk-HSM
- Einsatz für andere PKIs mit speziellen Anforderungen
 - Generische Import/Export-Schnittstellen (Billing, CRM)
 - Layout und Design austauschbar
 - Generische Schnittstellen zur Anbindung neuer Komponenten
- Einsatz im SigG-Kontext durch Verschlinkung

Anforderungen: Skalierbarkeit

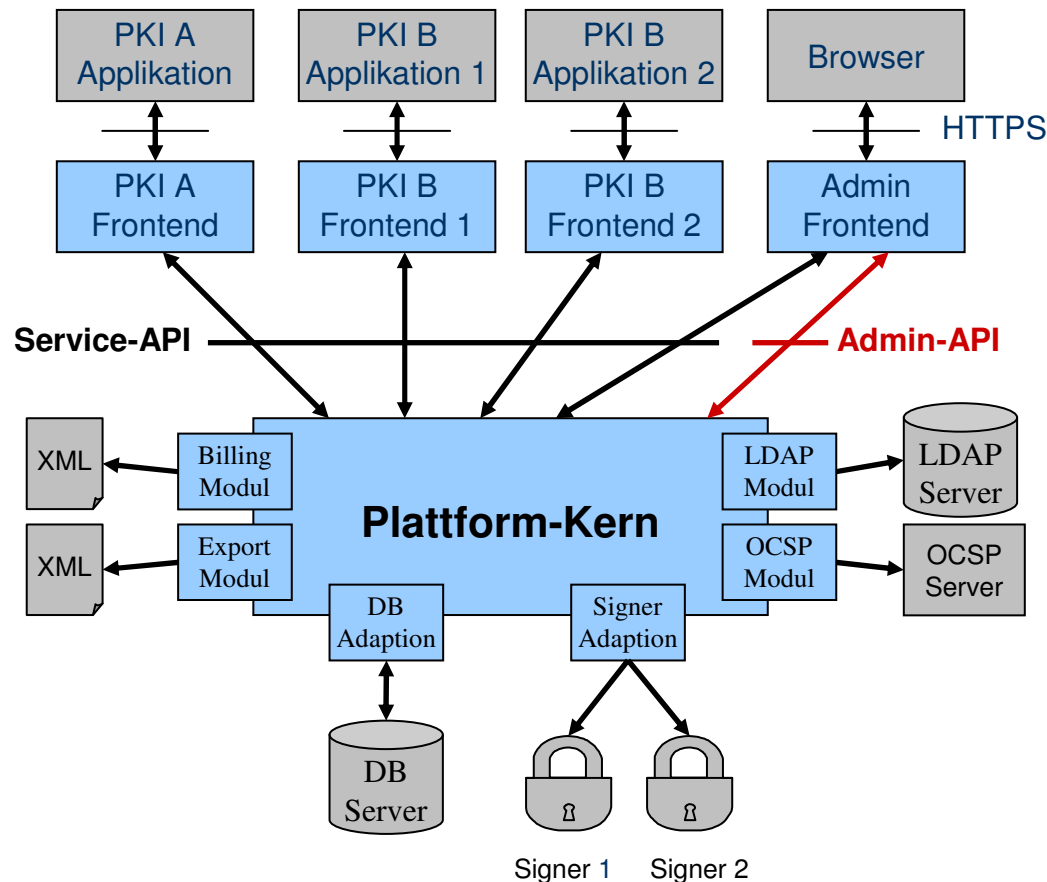


- Steigerung der Leistungsfähigkeit durch Hinzufügen von Ressourcen
 - Verteilbarkeit: Auslagerung von Komponenten auf separate Hardware
 - Interne *Schnitt-Stellen* frühzeitig vorsehen
 - Ohne Trennung auch kein Overhead für Remote-Anbindung
 - Redundanz: Mehrfache Installation einer Komponente mit Lastverteilung
 - „Zustandslose“ Komponenten
 - Standardprotokolle für HW-Lastverteilung
 - Kleine Arbeitspakete

Anforderungen: Administration

- Zentrale, mandantenübergreifende Administrator-Rolle
- Weitere Administrator-Rollen je Mandant
- Ein- und Ausblendung von Menüpunkten je nach Rechten des Nutzers
- Ein- und Ausblendung von Menüpunkten je nach installierten Modulen
- Zentrale Konfiguration, zentrales Logging, zentrale Überwachung
- SigG-Anforderungen
 - Ereignisprotokollierung signiert mit Zeitstempeldienst
d.h. geschützt vor Manipulationen (auch durch Administratoren)
 - Ansteuerung Langzeitarchiv
 - 4-Augenprinzip
- Verbindung verschiedener Installationen
 - Import/Export CA-Zertifikate und CRLs
 - Anmeldung mit gleichem RA-Zertifikat

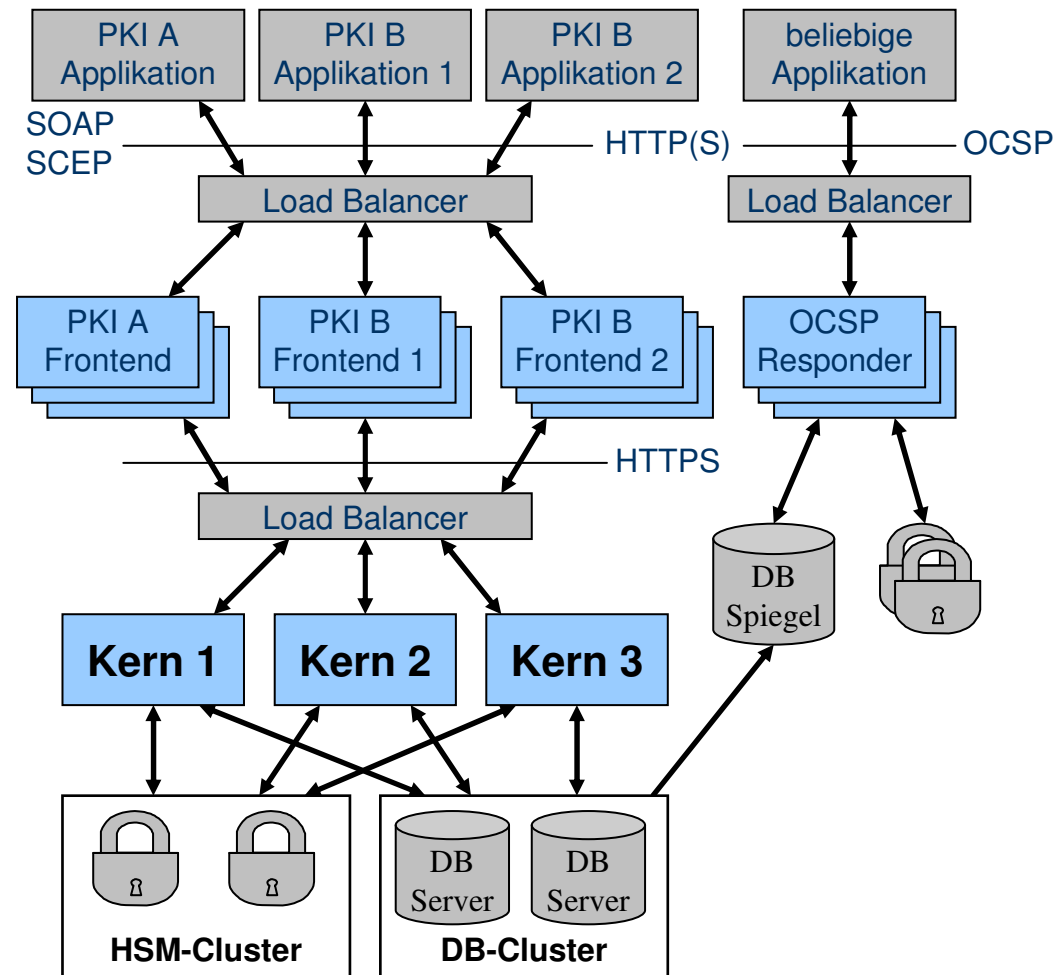
Architekturvorschlag: Komponenten und Module



- Übergreifende Funktionalität im Kern, PKI-Spezifisches in den Frontends
- Separate Schnittstelle für administrative Funktionen
- Schnittstellen durch Module und Adaptionsschichten abstrahiert
- Zentrale DB als einzige Persistenzkomponente
⇒ Installation, Administration, Skalierung, Backup usw. vereinfacht

Architekturvorschlag: Skalierbarkeit

- Protokolle über HTTP(S) wie OCSP, SCEP und SOAP erlauben schnelles Load-Balancing mit Standard-Komponenten
- Kern, Frontends und OCSP-Responder als Web-Applikationen realisiert
- Einheitliches Design
- DB-Spiegel und eigene Signer für OCSP-Responder zur gegenseitigen Entlastung



Vielen Dank für die Aufmerksamkeit!

Fragen?