# Vorlesung "Softwaretechnik"

# Projektbeispiel: elektronische Gesundheitskarte

Lutz Prechelt

Freie Universität Berlin, Institut für Informatik

- Einordnung
- Anforderungen 'eRezept'
  - funktionale, Leistungs-, Verfügbarkeits-, Sicherheits-
- techn. Ablauf, einige Details
- Beteiligte, Nutznießer und Konfliktlinien, Zeitverlauf

"Merke"-Hinweise zu:

- Domänen
- nichtfunktionale Anforderungen
- Kooperationsbedarf
- Projektrisiko

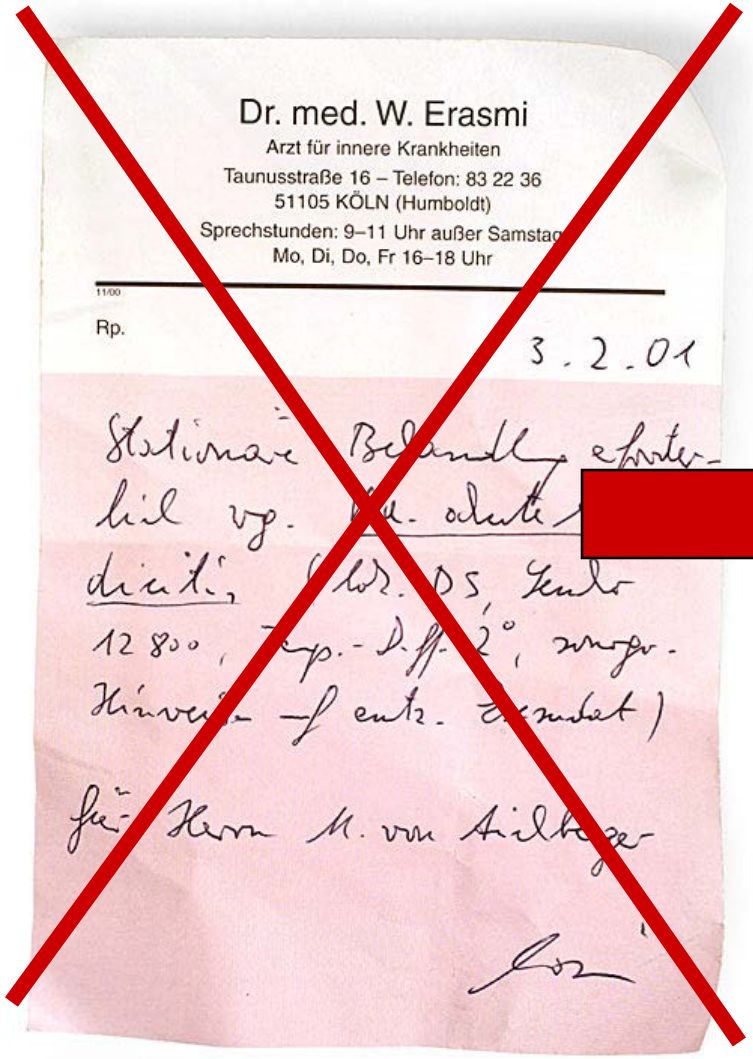
- Anhang: Digitale Signatur

# Lernziele

- Einige wichtige SWT-Begriffe kennen lernen
- Ein (vages) Gefühl für die Komplexität eines Großprojekts bekommen
- Quellen der Komplexität verstehen
  - Anwendungsdomäne
  - nichtfunktionale Anforderungen
- Ein paar Notationen mal gesehen haben

# Mal angenommen…

# Und jetzt modernisieren wir:


Dr. med. W. Erasmi
Arzt für innere Krankheiten
Taunusstraße 16 – Telefon: 83 22 36
51105 KÖLN (Humboldt)
Sprechstunden: 9–11 Uhr außer Samstag
Mo, Di, Do, Fr 16–18 Uhr

11/00

Rp.

3.2.01


Elektronische Gesundheitskarte (eGK)

# Anwendungen der eGK

1. **elektronisches Rezept** (eRezept, eVerordnung, Verordngsdatendienst, VODD):
- Arzt kann eRezept für eGK ausstellen
  - Aber nur ein *Arzt* kann dies tun
  - Evtl. unter Mitwirkung Arzthelfer
  - Evtl. ohne Vorlage der eGK (Nachfolgerezept, telefonisch!)
- Apotheker kann eRezept von eGK einlösen
  - Aber nur garantiert je einmal
  - Evtl. ohne Vorlage der eGK (Versandapotheke!)
- Patient kann eRezept löschen oder verbergen

2. **Weitere Anwendungen:**
- Notfalldaten
  - Blutgruppe, Allergien etc.
  - *Beschleunigt Rettung*
- Arztbrief
- Arzneimitteldokumentation
  - mit Interaktionsprüfung
  - *Vermeidet Medikamentunfälle*
- elektronische Patientenakte
  - kompl. medizinische Historie
  - *senkt Kosten (Doppelunters.!)*
  - *verbessert Versorgung individuell*
  - *verbessert Gesundheitssystem*
    - *Epidemiologie*
- Organspende-Ausweis, Patientenverfügung u.a.

# Hinweis: Zweck der eGK-Studie

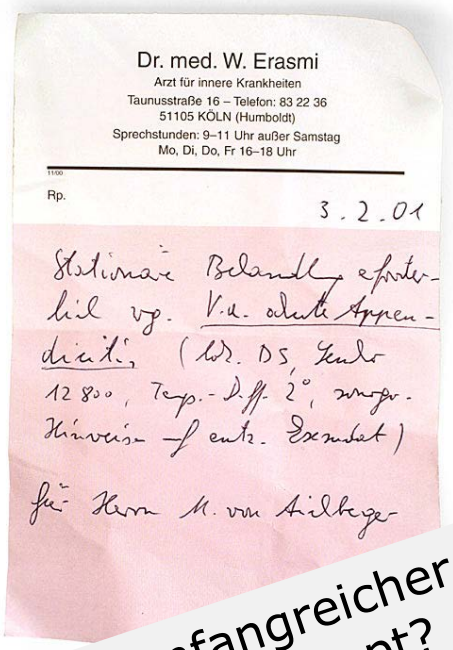
- Das eGK-Projekt ist derzeit im Gange
- Die zugehörigen Anforderungs- und Entwurfsdokumente sind (inzwischen: waren) öffentlich zugänglich
  - Konzeption: http://www.dimdi.de/static/de/ehealth/karte
    - "Rahmenarchitektur" (**RA**), "Lösungsarchitektur" (**LA**)
  - Umsetzung: http://www.gematik.de
  - (es hat inzwischen unzählige Änderungen gegeben!)
- Wir studieren die eGK, um die **Komplexität großer Softwareprojekte** zu **begreifen**
  - Achten Sie vor allem auf Art und Menge nötigen Wissens
    - nicht auf dessen Details.
  - Wir betrachten nur einen winzigen Ausschnitt des eGK-Projekts!
- Ich verweise später immer mal wieder auf dieses **Beispiel**

# Preisfrage: Datenstruktur eRezept

- Wie würden Sie die Klasse oder Klassen gestalten, die ein eRezept darstellen?

- Vielleicht etwa so?:

```
class eRezept {
    String  arzt;
    String  patient;
    String  arznei;
    int     packungsgroesse;
    String  einnahmeanleitung;
}
```


Dr. med. W. Erasmi
Arzt für innere Krankheiten
Taunusstraße 16 – Telefon: 83 22 36
51105 KÖLN (Humboldt)
Sprechstunden: 9–11 Uhr außer Samstag
Mo, Di, Do, Fr 16–18 Uhr

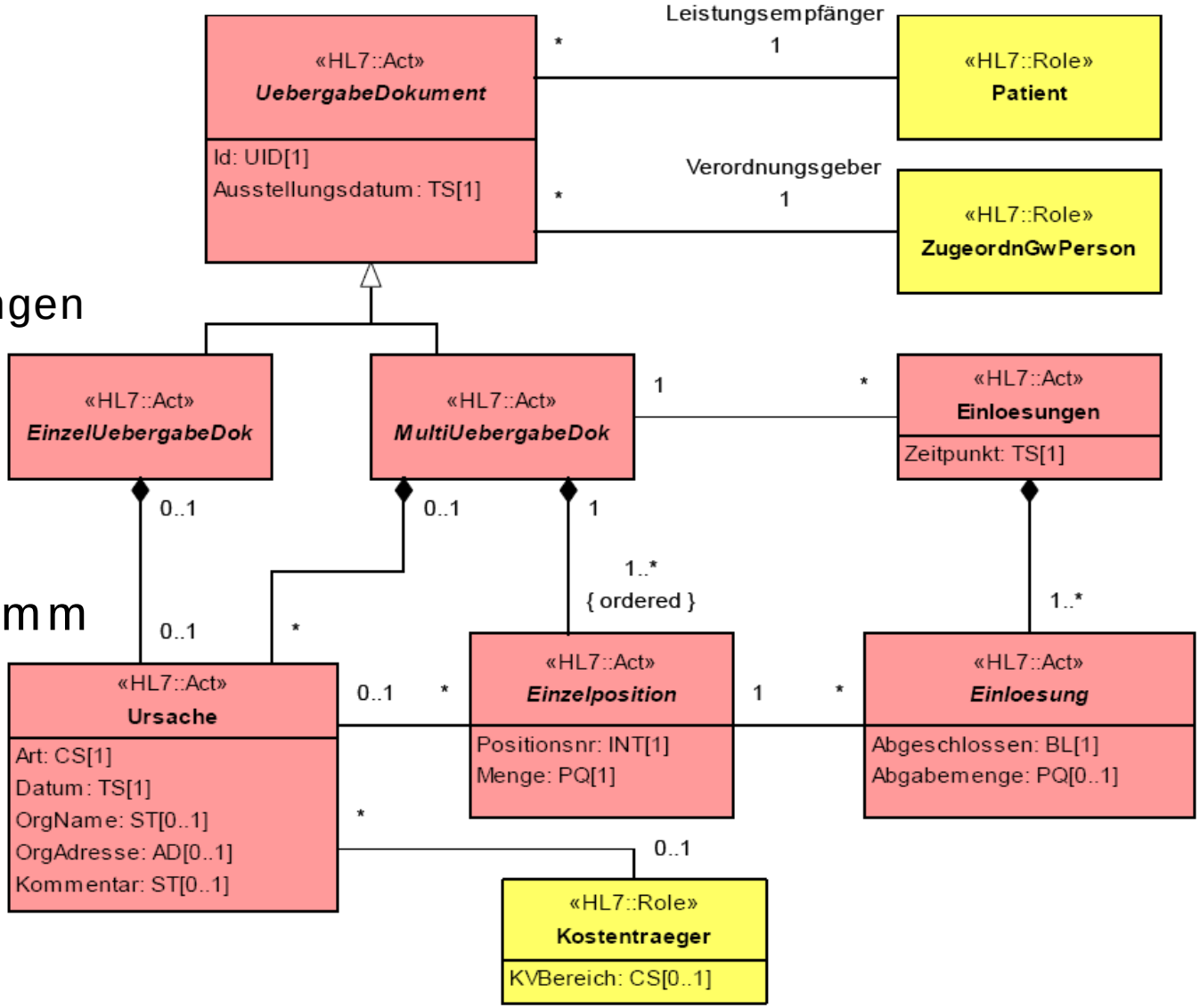
Rp.

3.2.01

Wieviel Prozent umfangreicher ist wohl das echte eRezept?

# Klassenmodell Übergabedokument

- RA 1.1 Informations-modell, S.41
  - 10 Klassen
  - 12 Beziehungen
- Notation: UML-Klassendiagramm (plus Farben)



# Umsetzungsphase: Klassenmodell eVerordnung

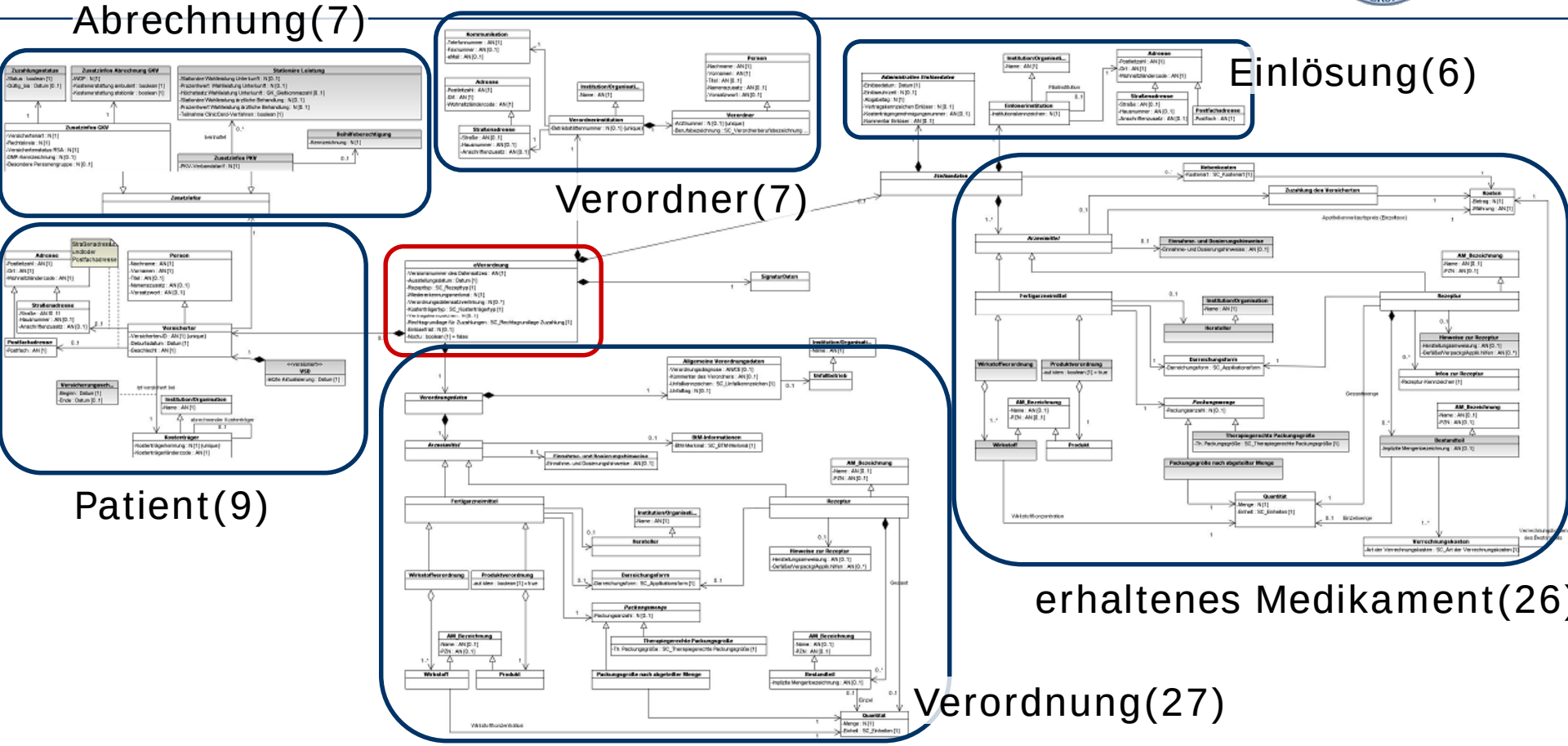



**Abbildung 8 – Infomodell VODM zu Release 2**

- gematik_VOD_Fachkonzept_VODM_V2_4_0.pdf, Seite 36
  - Erläuterung folgt auf Seiten 37-76
- Notation: UML-Klassendiagramm (implementierungsnah)

# Hilfe!!

- **Wo kommt diese enorme Komplexität her?**
- Von den Randbedingungen dieses Projekts, insbesondere
  - der existierenden Umgebung:
    - organisatorisch (wie das Gesundheitssystem arbeitet)
    - rechtlich (gesetzliche Notwendigkeiten)
    - nur wenig auch technisch (vorhandene DV-Systeme)
  - den Zielen (für eRezept aber eher weniger relevant):
    - technische Flexibilität (f. zusätzliche künftige Anwendungen)
    - internationale Harmonisierung (Standards!)
    - Bezahlbarkeit (Beibehaltung existierender Systeme) u.a.
- All dies zusammen definiert die **Anforderungen**
  - Diese muss man kennen und verstehen, um das eGK-System entwerfen zu können
- Sehen wir uns also ein paar Anforderungen an:

# Anforderungen global: zu beachtende Standards

Freie Universität Berlin

- Arten medizinischer Standards:
  - Klassifikationsstandards
    - ICD-10, ICD-O, ICF, OPS-301, PZN, ATC, G-DRG
    - unser Fall ist z.B. in ICD-10 wohl ein S76.2: S: Verletzungen, 76: des Oberschenkels, 2: Muskeln der Adduktorengruppe
    - siehe ICD-10 GM 2020 auf https://www.dimdi.de/static/de/klassifikationen/icd/icd-10-gm/kode-suche/htmlgm2020/
  - Terminologie-Standards (Nomenklaturen)
    - SNOMED (siehe z.B. RA 1.1. Standards, S.33), LOINC, MeSH, UMLS
  - Dokumentenformat-Standards
    - HL7 (siehe z.B. RA 1.1 Standards S.47), RIM, CDA, DICOM, xDT
- Weitere Arten relevanter Standards:
  - eGovernment-Standards
    - z.B. UML, XML Schema, J2EE, HTML, Messaging etc.
  - Sicherheitsstandards (https, SSL, AES, SHA-3, etc.)
  - Smartcard-Standards (CT-API, PC/SC2, etc.)

# Merke: Domänen sind kompliziert

Freie Universität Berlin

- Das Umfeld und der Anwendungsbereich, in dem ein SW-Projekt angesiedelt ist, heißt dessen Domäne
  - **Anwendungsdomäne** (problem domain, Problembereich): Wozu die Software dienen soll, plus dessen ganzes Umfeld
  - **Technische Domäne** (solution domain, Lösungsbereich): Mit welcher Technik die Software realisiert wird
  - (Sagt man nur "Domäne", ist meist die Anwendungsdomäne gemeint)

- Domänen weisen meist eine hohe Komplexität auf:
  - fachlich: Fachbegriffe, "selbstverständliche" Sachverhalte, Kultur, Standards, Verfahren/Prozesse, Rollen, Anforderungen
  - technisch: Sprachen, Technologien, Standards, Komponenten

- Sehr viel Wissen ist nötig

  ➔ Zusammenarbeit von Fachspezialisten und Technikspezialisten

# Anforderungen global: eGK Ausgangssituation

Freie Universität Berlin

- Es gibt Computersysteme bei den *Leistungserbringern*
  - Ärzten und Zahnärzten
  - Apotheken
  - Krankenhäusern
- Diese heißen *Primärsysteme*

- Es gibt Computersysteme bei den Krankenversicherern
  - gesetzliche Krankenversicherungen
  - private Krankenversicherungen
  - Berufsgenossenschaften
  - etc.
- Diese heißen *Backendsysteme*

- Es gibt genaue Vorgaben zur gesetzl. Krankenversich. (GKV)
  - Sozialgesetzbuch Teil 5 (SGB V) und ferner die PKV

# Anforderungen global: eGK Zielvorstellung

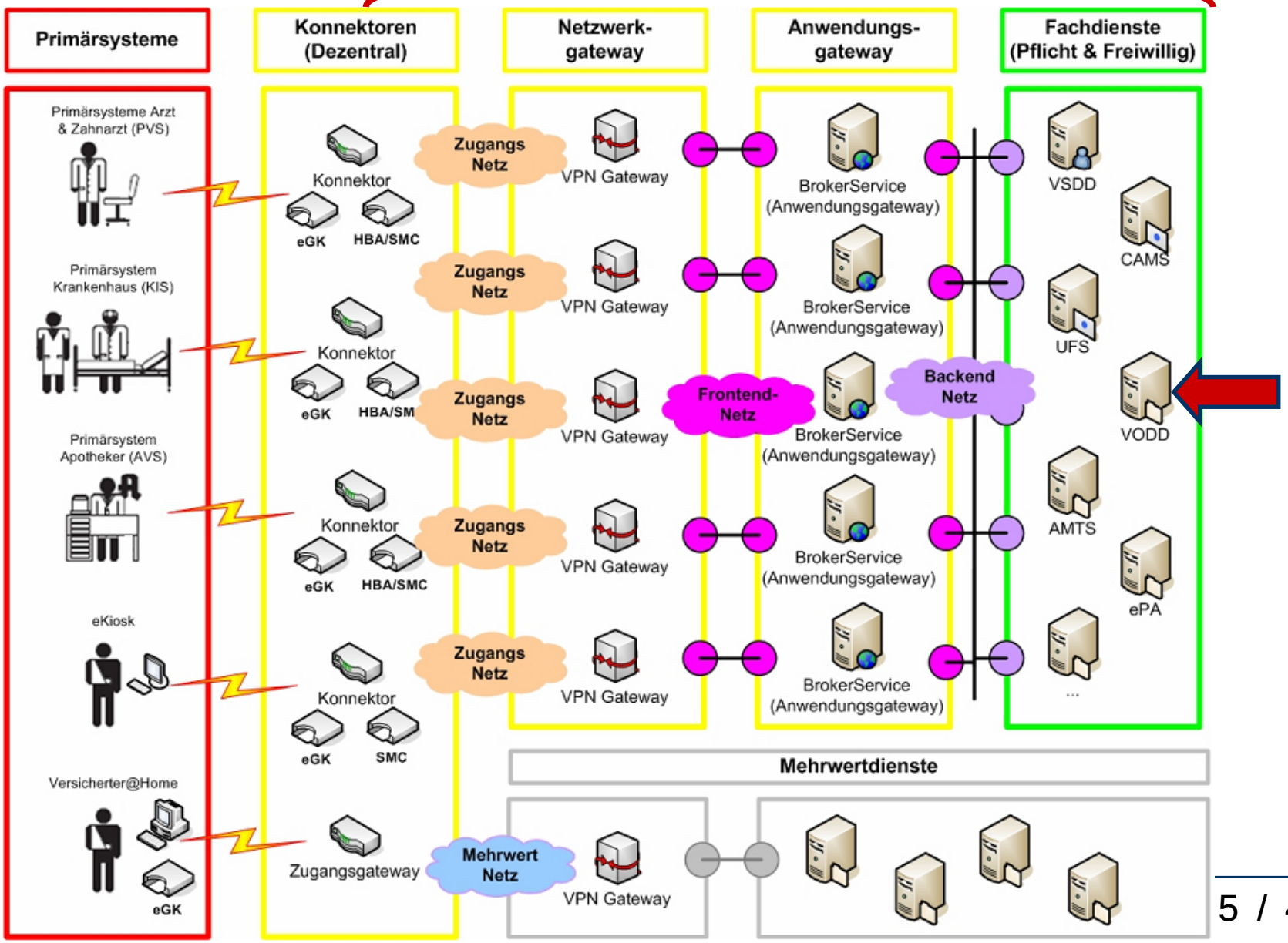
Freie Universität Berlin

- Das Projekt soll die effiziente Integration (Zusammenarbeit) der existierenden Systeme ermöglichen
  - unter Beachtung aller gesetzlicher Vorgaben
  - unter starker Nutzung medizinischer u. technischer Standards

- Dazu stellt es bereit:
  - Zur Integration: *Telematik-Infrastruktur*
    - Kommunikationsplattform, Basisdienste (z.B. Zeitstempel)
    - eigentliche Anwendungen (zur Erzwingung der Regeleinhaltung)
  - Für Leistungserbringer (Ärzte etc.): *bIT4health Connector*
    - f. sicheren Zugang der Primärsysteme zur Telematik-Infrastruktur
  - Für Patienten: eGK
    - zur Identifikation und zur Wahrung ihrer Informationshoheit
    - abgesichert durch Verschlüsselung und digitale Unterschriften
  - Ferner: Kartenterminals, SMC (Institutionsausweis), Health Professional Card (HPC, *dt.:* Heilberufsausweis, HBA)

# Zielvorstellung: Grobsicht auf Telematik-Infrastruktur

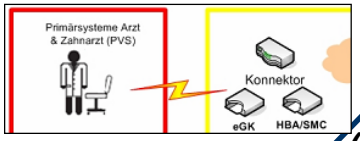
Freie Universität Berlin



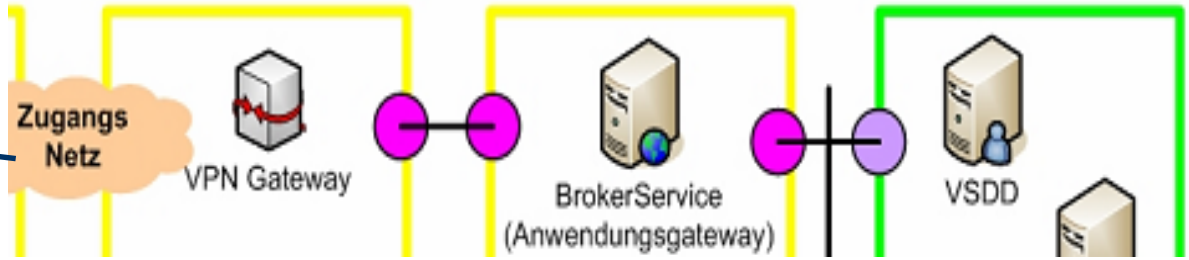

gematik_GA_Gesamtarchitektur_V1_1_0.pdf, S.26

# Zielvorstellung: Situation in der Arztpraxis

PVS




Konnektor (hier: Siemens)




zentrale Telematik-Infrastruktur

Arztpraxis

Kartenterminal (hier: Jarltech)

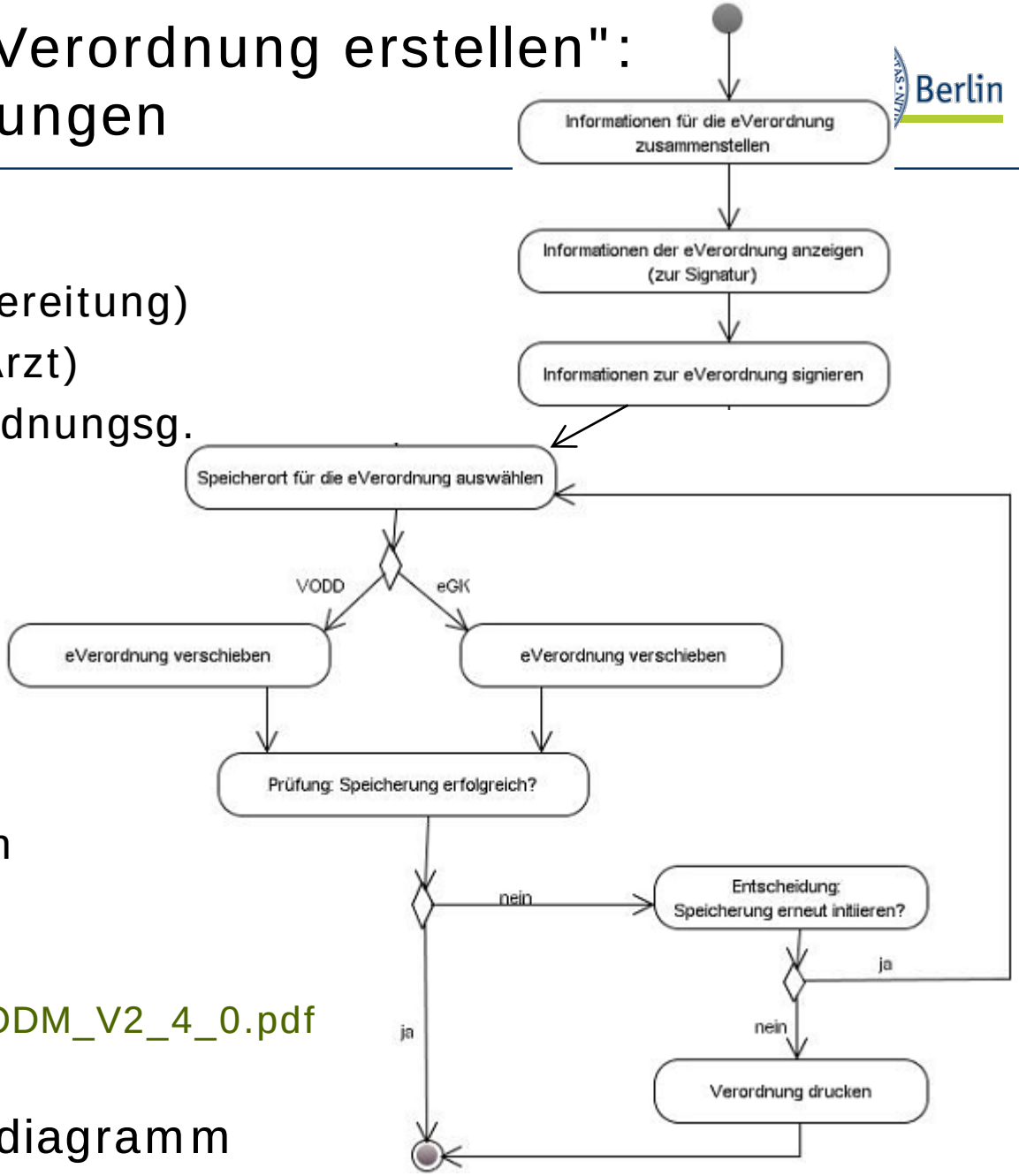

eGK (Patient)


HBA (Arzt)

Ist diese Struktur eine gute Idee?

# 1. Anforderungen "eVerordnung erstellen": funktionale Anforderungen

- Akteure:
  - Patient (nur bei Vorbereitung)
  - Verordnungsgeber (Arzt)
  - Mitarbeiter des Verordnungsg.
- Vorher nötig:
  - Bereitstellung eGK durch Patient
  - Authentisierung Arzt/HBA
- Als separater Ablauf:
  - eVerordnung einlösen (Apotheke)




gematik_VOD_Fachkonzept_VODM_V2_4_0.pdf
Seite 24-25

Notation: UML-Aktivitätsdiagramm

# 2. Anforderungen "eVerordnung erstellen": funktionale Anford. als Use-Case (1)

Freie Universität Berlin

| UC_23 - eVerordnung_erstellen_Abschnitt_3 | |
|---|---|
| Beschreibung | Die Informationen für eine eVerordnung werden zusammengestellt und signiert. Die eVerordnung wird auf eGK oder VODD verschoben. |
| Anwendungsumfeld | Institution des Akteurs Arzt (z. B. Arztpraxis, Krankenhaus) |
| Vorbedingungen | • Rechtmäßige Nutzung der eGK ist überprüft.<br>• technisch nutzbare eGK liegt vor (gemäß [gemFK_CMSeGK_Nutz]).<br>• Zugriffsauthentisierung durch Akteure ist erfolgt. |
| Auslöser | Eine eVerordnung soll erstellt werden. |
| Eingangsdaten | VSD_eGK / VOD_AM / Signaturdaten |
| Ergebnisse | Die eVerordnung wurde auf VODD oder eGK gespeichert. |
| Nachbedingungen | Die eVerordnung steht auf VODD oder eGK bereit zur weiteren Verwendung. |
| Beteiligte Akteure | - Arzt<br>- Mitarbeiter medizinische Institution]: |
| Geschäftsobjekte | eVerordnung |
| Standardablauf | 1 [Arzt oder Mitarbeiter medizinische Institution]:Informationen für die eVerordnung zusammenstellen<br>2 [Arzt]:Informationen der eVerordnung anzeigen (zur Signatur)<br>3 [Arzt]:Informationen der eVerordnung signieren<br>4 [Arzt oder Mitarbeiter medizinische Institution]:Speichern für die |

# 2. Anforderungen "eVerordnung erstellen": funktionale Anford. als Use-Case (2)

Freie Universität Berlin

| | |
|---|---|
| | 3 [Arzt]:Informationen der eVerordnung signieren<br>4 [Arzt oder Mitarbeiter medizinische Institution]:Speicherort für die eVerordnung auswählen: VODD oder eGK<br>5 [Arzt oder Mitarbeiter medizinische Institution]:eVerordnung verschieben<br>6 [Arzt oder Mitarbeiter medizinische Institution]:Prüfung: Speicherung erfolgreich? – JA |
| Varianten | Keine |
| Sonderfälle | 6a. [-]:Prüfung: Speicherung erfolgreich? - NEIN<br>6a.1 [-]:Fehlermeldung #1 oder #2<br>6a.2 [Arzt oder Mitarbeiter medizinische Institution]:Entscheidung: Speicherung erneut initiieren? - JA<br>6a.3 [-]:weiter mit Schritt4<br>6b [-]:Prüfung: Speicherung erfolgreich? - NEIN<br>6b.1 [-]:Fehlermeldung #1 oder #2<br>6b.2 [Arzt oder Mitarbeiter medizinische Institution]:Entscheidung: Speicherung erneut initiieren? - NEIN<br>6b.3 [-]:Verordnung drucken<br>6b.4 [-]:Ende des Anwendungsfalls |
| Fachliche Fehlermeldungen | #1 Speicherung auf VODD nicht möglich<br>#2 Speicherung auf eGK nicht möglich |

# 3. Nichtfunktionale Anforderungen: Mengengerüst

Freie Universität Berlin

- Anzahl zu verwaltender eGK:
  - 82 Millionen Versicherte
  - ➔ Herausforderung für Herstellungs- und Verteilungskapazität
- Anzahl zu verwaltender HBA:
  - 130.000 Ärzte (plus 150.000 in Krankenhäusern),
    56.000 Zahnärzte, 46.000 Apotheker,
    1,7 Mio sonstige Heilberufler (Optiker, Masseure, u.v.a.m.)
  - ➔ Herausforderung f. Herstellungskapazität bei Kartenterminals
- Anzahl eRezepte:
  - 335.000.000 Rezepte jährlich (mit 570.000.000 Verordnungen)
  - ➔ erhebliche Datenmengen und hoher Verarbeitungsbedarf
  - ➔ verlangt verteiltes System mit hohem Durchsatz

gematik_GA_Gesamtarchitektur_V1_1_0.pdf, Seite 199-200

# 4. Nichtfunktionale Anforderungen: Verfügbarkeit

Freie Universität Berlin

Verfügbarkeit ist der Prozentsatz der Zeit, in dem ein System nicht ausgefallen ist (also bereit)

- RA definiert Verfügbarkeitsklassen f. alle Teilfunktionen
  - V1: hochverfügbar: 99.999%, Ausfall nicht tolerabel
    - d.h. max. 5 Minuten Nichtverfügbarkeit pro Jahr
  - V2: zeitkritisch: 99.99%,
    max. 30 Min. Ausfall und Ausweichlösungen
    - d.h. max. 50 Minuten Nichtverfügbarkeit pro Jahr
  - V3: weniger zeitkritisch: 99.9%, max. 8 Std. Ausfall
    - d.h. max. 9 Stunden Nichtverfügbarkeit pro Jahr

➔ Verlangt Redundanz in der HW und Fail-Over

Dokument RA 1.1 nichtfunktionale Anforderungen

# 5. Nichtfunktionale Anforderungen: Sicherheitsanforderungen

Freie Universität Berlin

| | | | |
|---|---|---|---|
| AS_VOD_AM_01_05<br>AS_VOD_AM_02_07<br>AS_VOD_AM_01_07<br>AS_VOD_AM_02_10<br>AS_VOD_AM_03_08<br>AS_VOD_AM_04_08 | Duplikatserstellung | Bei der Speicherung der eVerordnung auf der eGK MUSS sichergestellt werden, dass kein Duplikat der eVerordnung in der Telematikinfrastruktur erstellt wird. | MUSS |
| AS_VOD_AM_02_02 | Verschlüsselung | Eine eVerordnung MUSS in verschlüsselter Form auf den VODD übertragen und dort gespeichert sein. Den Schlüssel zum Entschlüsseln der eVerordnungen besitzt der Versicherte. | MUSS |
| AS_VOD_AM_02_08 | Pseudonymisierung des personenbezogenen Speicherortes der eVerordnung | Der Speicherort der Verordnung innerhalb der Telematikinfrastruktur DARF NICHT für Unberechtigte erkennbar sein. (Pseudonymisierung) | DARF NICHT |

- und diverse andere

gematik_VOD_Fachkonzept_VODM_V2_4_0.pdf, Seite 84-85

# Merke: Nichtfunktionale Anforderungn sind oft dominant

Freie Universität Berlin

- Man ahnt bereits jetzt:
  - Die eigentliche Funktion des eGK-Systems ist zwar recht simpel
  - aber durch die Sicherheitsanforderungen wird es sehr kompliziert

- Das ist typisch für größere Softwaresysteme:
  - Die Komplexität entsteht vor allem aus den nichtfunktionalen Anforderungen
  - Je nach Einzelfall z.B.:

| | (bei eGK) |
|---|---|
| Sicherheitsanforderungen (security) | + + |
| Leistungsanforderungen (Durchsatz oder Skalierung) | o |
| Verteilung, Mobilität | +, - |
| Zuverlässigkeitsanforderungen | + + |
| Benutzbarkeit | ?? |

# 5. Nichtfunktionale Anforderungen: Sicherheitsanforderungen (3)

Freie Universität Berlin

- Die Nichteinhaltung von Sicherheitsanforderungen ist oft sogar ein Gesetzesverstoß! Beispiel:

§ 291 a Abs. 4 Satz 1 SGB V: (www.sozialgesetzbuch.de)

Zum Zwecke des Erhebens, Verarbeitens oder Nutzens mittels der elektronischen Gesundheitskarte **dürfen**, soweit es zur Versorgung der Versicherten erforderlich ist, **auf Daten**:
**1.** nach Abs. 2 Satz 1 Nr. 1 *[eRezepte]* ausschließlich a) Ärzte, b) Zahnärzte, c) Apotheker, [...] Apothekerassist., Apothekenassistenten, d) Personen, die aa) bei den unter Buchstabe a bis c Genannten oder bb) in einem Krankenhaus als berufsmäßige Gehilfen oder zur Vorbereitung auf den Beruf tätig sind, soweit dies im Rahmen der von ihnen zulässigerweise zu erledigenden Tätigkeiten erforderlich ist und der Zugriff unter Aufsicht der in Buchstabe a bis c Genannten erfolgt, e) sonstige Erbringer ärztlich verordneter Leistungen,
**2.** nach Absatz 3 Satz 1 Nr. 1 bis 5 *[restl. Daten]* ausschließlich **[**a) bis d) wie vor**]** e) nach Absatz 3 Satz 1 Nr. 1 in Notfällen auch Angehörige eines anderen Heilberufs, der für die Berufsausübung oder die Führung der Berufsbezeichnung eine staatlich geregelte Ausbildung erfordert, f) Psychotherapeuten
**zugreifen**.

# Merke: Domänenwissen ist wichtig

Freie Universität Berlin

Wichtige Erkenntnis:

- In vielen SW-Projekten kann man die Anforderungen nur verstehen, wenn man Wissen im betroffenen **fachlichen** Bereich ("**Anwendungsdomäne**") hat
  - (kennen Sie z.B. den Unterschied zwischen Apotheker<u>r</u>assistenten und Apotheke<u>n</u>assistenten?)
  - Und ohne Verständnis der **Anforderungen** kann man offensichtlich kaum für brauchbare Ergebnisse garantieren…

- Es gibt viele solche fachlichen Bereiche
- In mindestens einem davon sollte man sich auskennen!

Wieviel Prozent dieser Domäne kennen Sie jetzt?

# Äh, wo waren wir doch gleich?

Schritte für "eVerordnung erstellen" (laut Use Case):

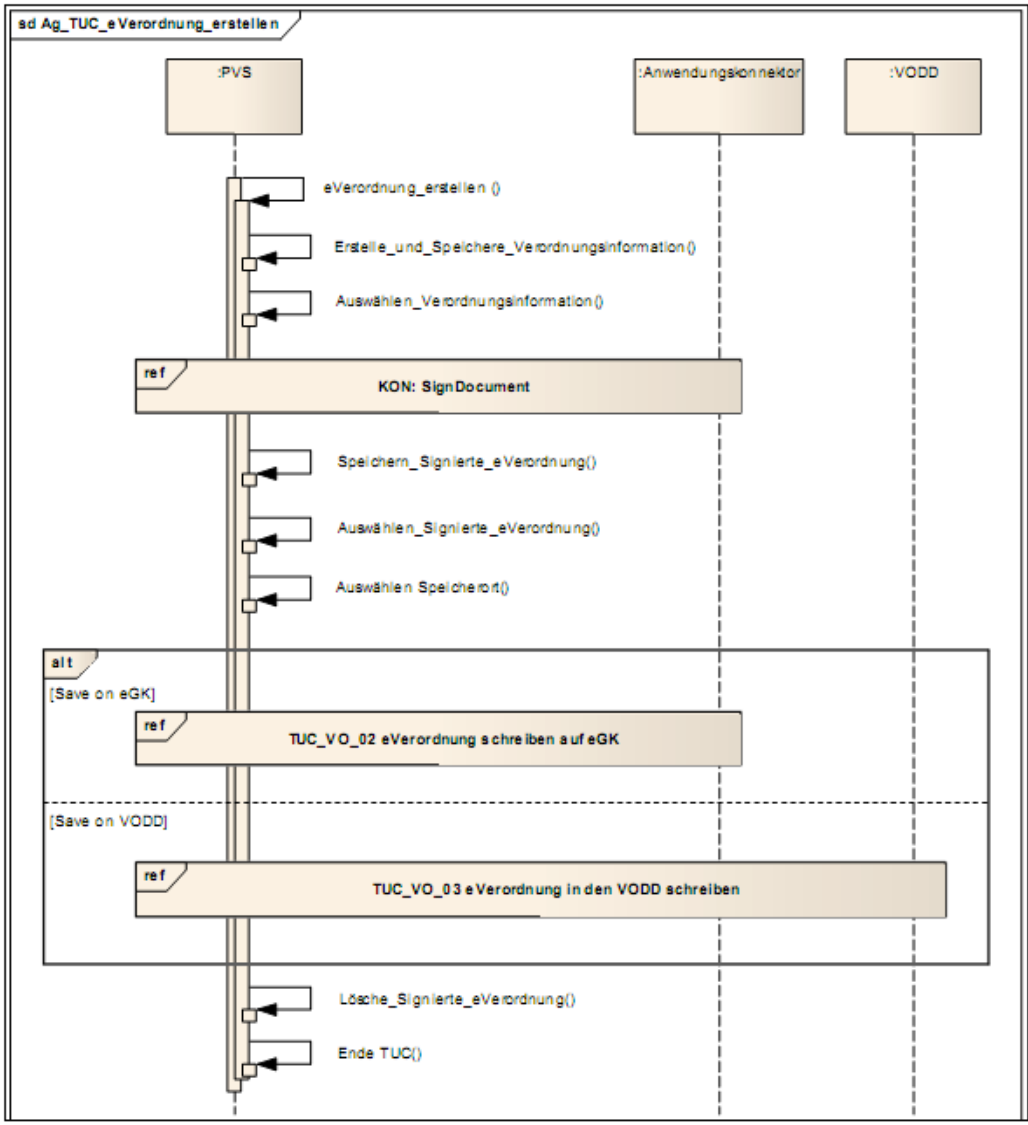
- 1 Informationen für die eVerordnung zusammenstellen (PVS)
- 2 anzeigen zur Signatur (PVS)
- 3 eVerordnung signieren (PVS, Konnektor, HBA)
- 4 Speicherort auswählen: VODD oder eGK (PVS)
- 5a eVerordnung auf eGK schreiben *oder*
- 5b eVerordnung in VODD schreiben

  - VODD: Verordnungsdatendienst
  - oben hieß das Schreiben "Verschieben"

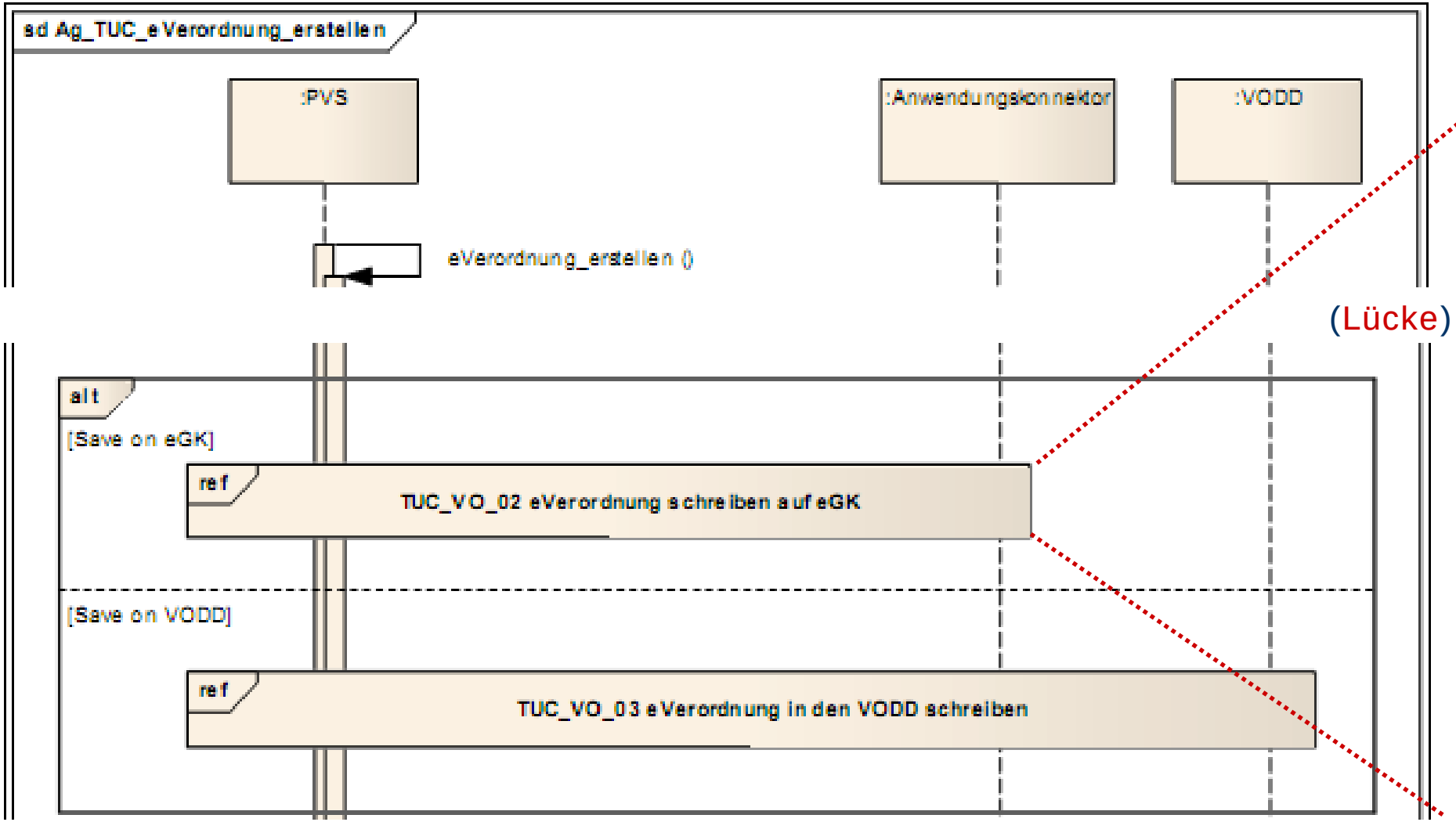
# Entwurf und Realisierung: Ag_TUC *eVerordnung erstellen*

- Notation: UML-Sequenzdiagramm
  - zu lesen von oben nach unten (Zeitablauf)
  - beschreibt wie verschiedene Objekte (senkrechte Bahnen) interagieren (Pfeile)
    - Hier: PVS, Konnektor, VODD
  - kann andere Sequenzdiagramme referenzieren (horiz. Kästen)

- gematik_VOD_Facharchitektur_VODM_V1_3_0.pdf, Seite 16



# Entwurf und Realisierung: Ag_TUC *eVerordnung erstellen* (teilw.)

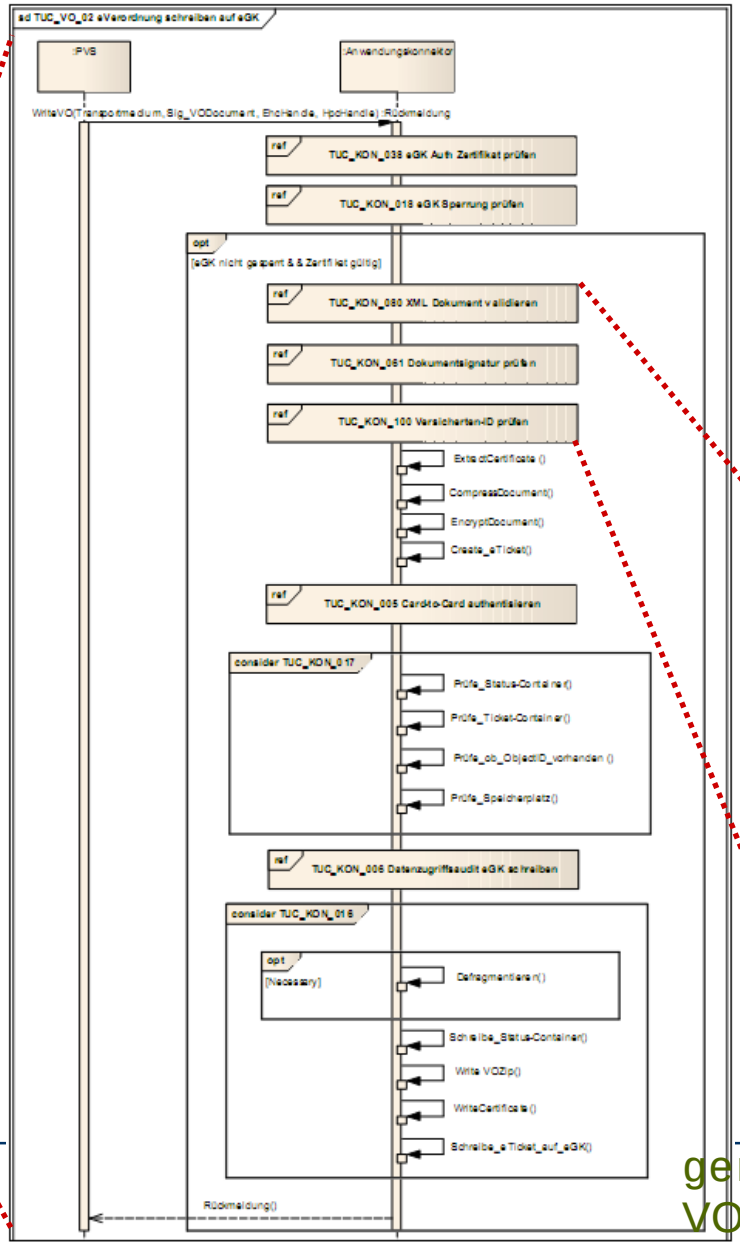



(Lücke)

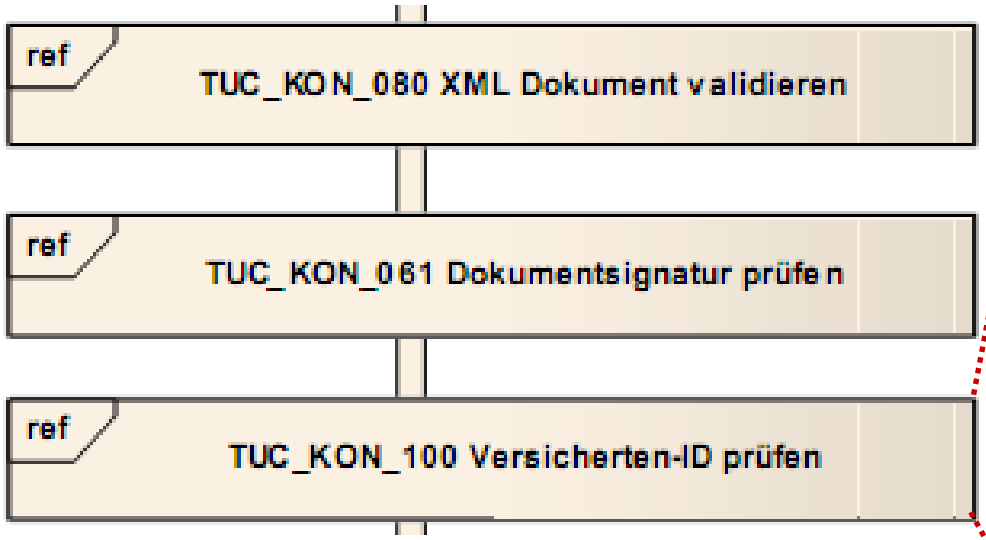
gematik_VOD_Facharchitektur_VODM_V1_3_0.pdf, Seite 16

(Lücke)

# Entwurf und Realisierung: TUC VO 02 *eVerord. schreiben auf eGK*



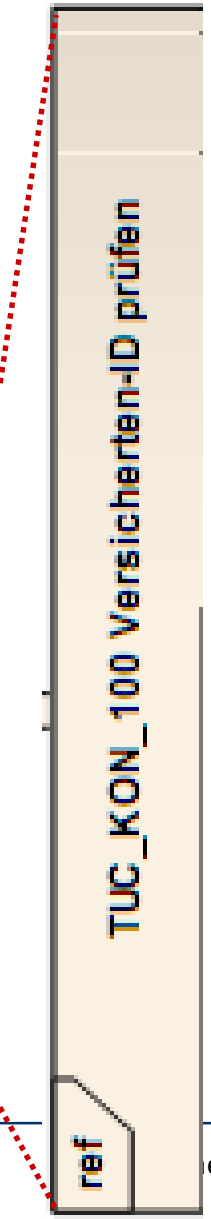

- TUC: Technical Use Case
- VO: Fachanwendg. eVerordnung
- Es gibt ferner: TUC VO 03 bis 11 für lesen, löschen, einlösen, Speicher bereinigen
  - von/auf eGK
  - von/auf VODD (Verordnungsdatendienst)




gematik_VOD_Facharchitektur_VODM_V1_3_0.pdf, Seite 18 [2]

# Entwurf und Realisierung: TUC KON 100 *Versicherten-ID prüfen*

Freie Universität Berlin




| Element | Beschreibung |
|---|---|
| Name | TUC_KON_100 „Versicherten-ID prüfen“ |
| Beschreibung | Es wird geprüft, ob die Versicherten-ID auf der eGK mit der Versicherten-ID in einem XML-Dokument einer Fachanwendung übereinstimmt (Verordnung, Notfalldaten,...). |
| Anwendungsumfeld | Dieser Use Case wird aufgerufen, um vor dem Schreiben eines Dokumentes auf die eGK zu prüfen, ob es sich um die Karte des Versicherten handelt, für den das Dokument bestimmt ist. |
| Auslöser | Fachanwendung im Konnektor ruft diesen Use Case auf. |
| Eingangsdaten / Vorbedingungen | • Versicherten-ID aus dem XML-Dokument oder XML-Dokument und Position, an der die Versicherten-ID zu finden ist (z. B. als XPath)<br>• eGK (Handle oder Objektreferenz) |
| Komponenten | Konnektor, eGK |
| Geschäftsobjekte | Notfalldaten, Verordnung u. ä., abhängig vom fachlichen Anwendungsfall. |
| Ausgangsdaten/ Nachbedingungen | Versicherten-ID der eGK und des Dokumentes stimmen überein: ja / nein |
| Referenzen | |
| Standardablauf | 1. Der Konnektor liest das Authentisierungszertifikat der eGK und extrahiert die Versicherten-ID aus dem Subject Distinguished Name des Zertifikats.<br>2. Es wird geprüft, ob die Versicherten-ID mit der im Dokument angegebenen übereinstimmt. |

# Fragen über Fragen

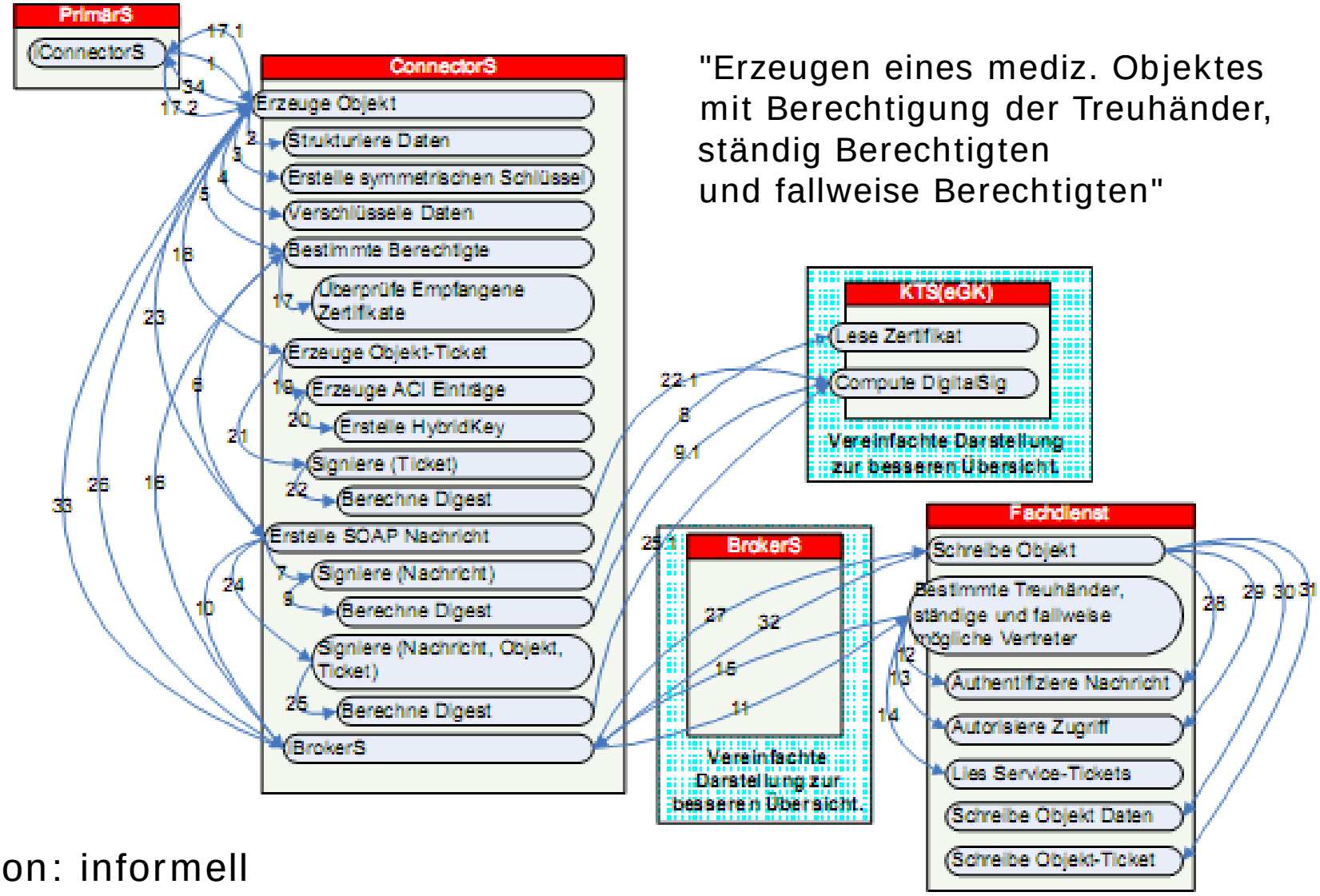
- So könnte man noch lange weiterbohren
- An fast jeder Stelle findet sich hohe Komplexität:
  - Was ist ein Zertifikat?
  - Wie kommuniziert der Konnektor mit dem Kartenterminal?
    - Das beansprucht ein komplettes, separates Dokument
  - Wie kommuniziert das Kartenterminal mit der eGK?
    - Das beansprucht ein komplettes, separates Dokument
  - Wie ist das eVerordnungs-XML-Dokument aufgebaut?
    - Warum ist das überhaupt XML?

- …und dabei hat ja diese ganze Operation mit der Telematik-Infrastruktur (VODD) nicht einmal zu tun!
  - es geht ja nur um Konnektor und eGK

- Es folgen ein paar willkürliche, zusammenhanglose Einzelheiten
  - um eine vage Vorstellung von der Vielschichtigkeit zu geben

Warum ist das alles so komplex?

# Detail: Kommunikationsmuster für einzelne Operationen (Beispiel)

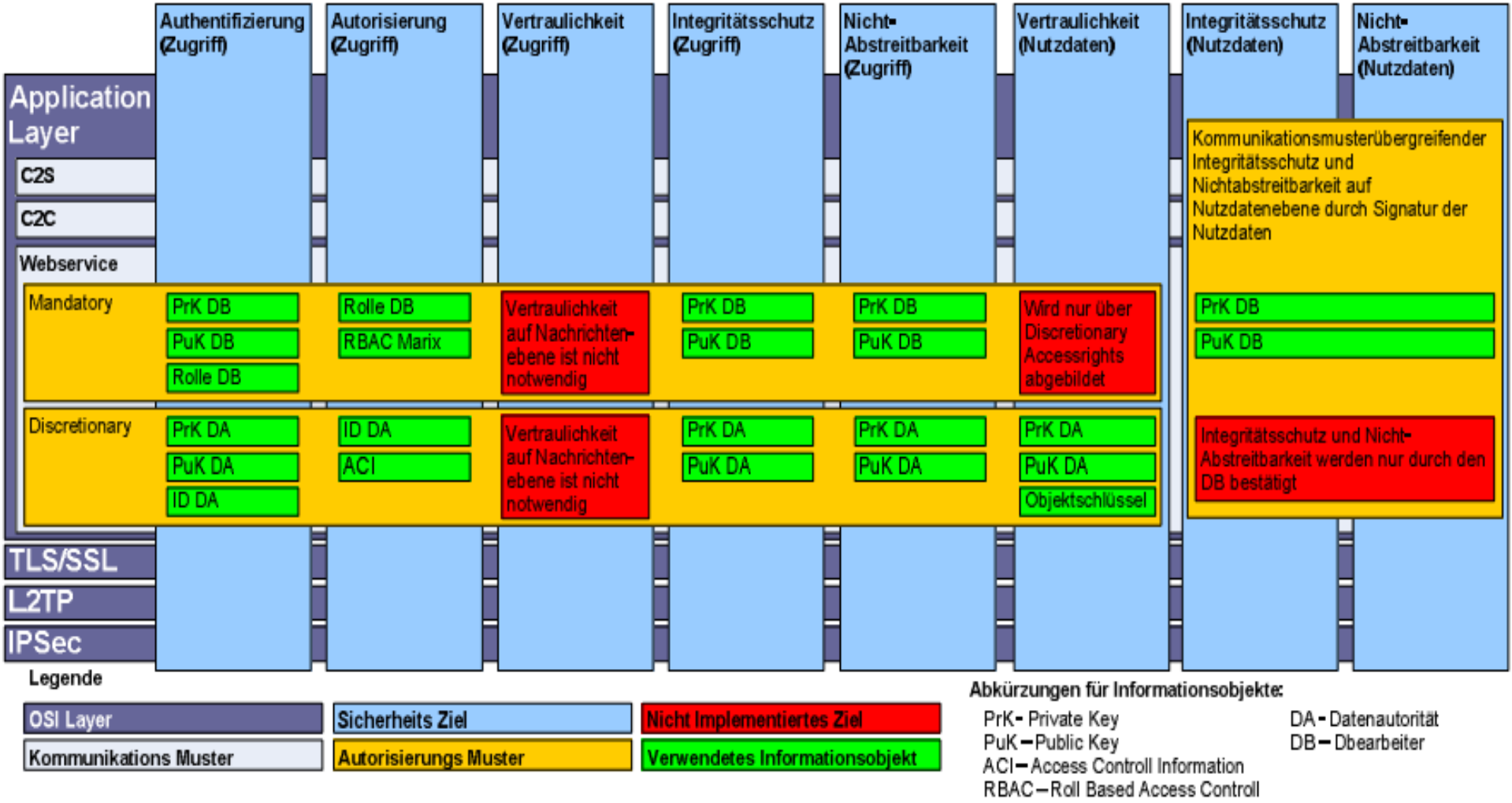



"Erzeugen eines mediz. Objektes mit Berechtigung der Treuhänder, ständig Berechtigten und fallweise Berechtigten"

Notation: informell

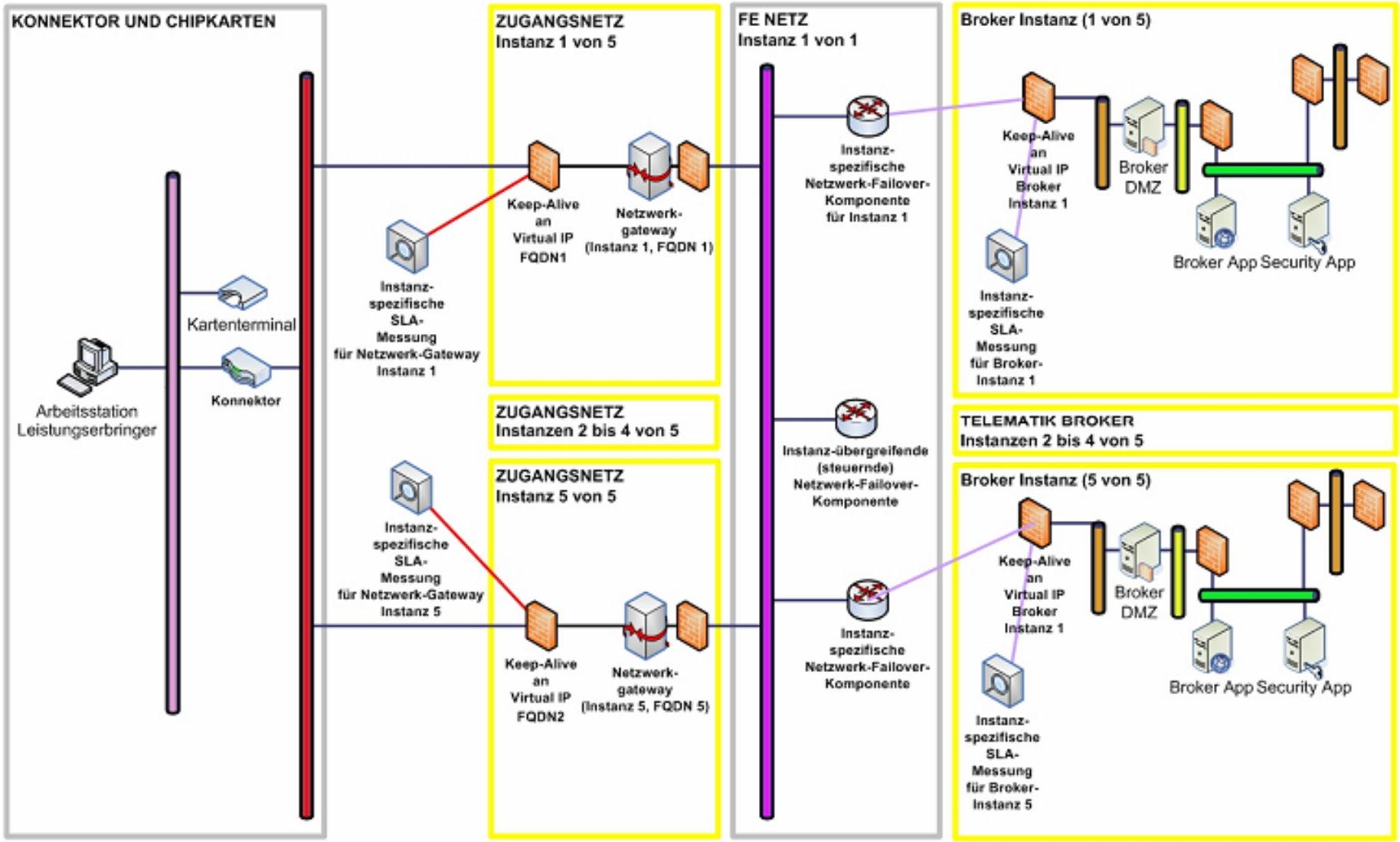
# Detail: Informationsobjekte f. d. Sicherheitsziele bei Webservice-Kommunikation

Freie Universität Berlin




Notation: informell

# Detail: Fail-Over-Architektur



# "Dokumentenlandkarte" (ohne RA + LA)

Freie Universität Berlin

Release 2.3.2 - VSD und VOD – Online - Feldtest

gematik Stand: 10.01.08

298

109

247

317

**Anforderungen an den Betrieb**
- Richtlinie für den Betrieb der Gesundheitstelematik — V1.0.0

**Sicherheitsanforderungen**
- Übergreifendes Sicherheitskonzept — V2.1.0
- Verwendung kryptographischer Algorithmen in der TI — V1.0.0

**Architekturanforderungen**
- Gesamtarchitektur — V1.1.0
- Anforderungen zur Unterstützung der Wartungs- und Änderungsproz. — V1.0.0

**Unterstützung der Testregionen**
- Gesamtkonzept Testverfahren — V1.1.0
- Richtlinie Feldtest 10.000 — V2.0.0
- Schulungskonzept — V2.0.0

## Fachanwendungen

**Fachkonzepte**
- Versichertenstammdaten-management (VSDM) — V1.0.0
- Verordnungsdatenmanagement (VODM) — V2.4.0
- Daten für die Notfallversorgung (NFDM) — V1.3.0

**Fachkonzepte**
- Kartenmanagement der eGK — V1.1.0
- Technische Nutzbarkeit der eGK — V1.0.0
- Beschreibung der zulässigen PIN- und PUK-Verfahren für die eGK — V1.1.0

**PKI für CV-Zertifikate**
- Grobkonzept — V1.2.0
- Registrierung einer CVC-CA der zweiten Ebene — V1.4.0

**PKI für X.509-Zertifikate**
- Grobkonzept — V1.2.0
- Festlegung einer einheitlichen X.509-Zertifikatsinfrastruktur — V1.0.0
- Registrierung eines Trust Service Provider (TSP) — V1.1.0
- Certificate Policy – Gemeinsame Zertifizierungs-Richtlinie für Teilnehmer der gematik-TSL zur Herausgabe von X.509-ENC/AUT/OSIG-Zertifikaten — V1.2.0
- Verzeichnisdienstkonzept — V1.1.0
- Konzeption u. Registrierungs-anforderungen für die TCL — V1.1.0
- X.509-Zertifikate der SMC Typ B — V1.2.0
- Signieren von Konfigurationsdaten und Software — V1.0.0
- System und Service Level Monitoring — V1.0.0
- Zulassungsverfahren für die Erstellung der krypt. Identität
  - Konnektor — V1.1.0
  - Server und Fachdienste — V1.0.0

## Spezifikationen

**Facharchitektur**
- Versichertenstammdatenmanagement (VSDM) — V2.2.0
- Verordnungsdatenmanagement (VODM) — V1.3.0
- Daten für die Notfallversorgung (NFDM) Offline — V1.4.0

**Facharchitektur**
- Speicherstrukturen der eGK — V1.4.0

**Kartenmanagement eGK**
- Facharchitektur Kartenmanagem. — V1.4.0
- eGK Schnittstellenspez. Card Communication Service CCS — V1.4.0
- eGK Schnittstellenspez. UFS — V1.4.0
- Übergabeschnittstelle für die Produktion der eGK — V1.4.0
- Personalisierung kryptographischer Daten der eGK — V1.0.0

**Die Spezifikation der eGK**
- Teil 1: Kommandos, Algorithmen und Funktionen der Betriebssystemplattform — V2.0.0
- Teil 2: Anwendungen und anwendungsspezifische Strukturen — V2.1.0
- Teil 3: Äußere Gestaltung — V2.1.0

**X.509- und CV-Zertifikate**
- Festlegungen zu den X.509-Zertifikaten der Versicherten — V1.4.0
- Verw. von Zertifikaten in der TI — V1.0.0
- Festlegungen zu den X.509-Komponentenzertifikaten
  - Konnektor — V1.1.0
  - Server und Fachdienste — V1.1.0

**Architekturfestlegungen**
- Spezifikation Fehlerbehandlung — V1.3.0
- Spezifikation TelematikTransport-Details — V1.4.0
- Spezifikation Ticketservice — V1.2.0
- Spezifikation Versionsnummern in Schnittstellen und Software — V1.1.0

**Infrastrukturdienste**
- Netzwerkspezifikation — V1.5.0
- Netzwerksicherheit — V1.1.0
- Broker-Service — V1.1.0
- Registrierungsdienst (SDS) — V1.1.0
- Namens-/Lokalisierungsdienst — V1.2.0
- Zeitdienst — V1.2.0
- Sytem und Service Level Monitoring — V1.3.0

**Konnektor**
- Konnektorspezifikation — V2.4.0

**SICCT-Kartenterminal**
- Spezifikation — V1.20

**eHealth Kartenterminal**
- Spezifikation eHealth-Karten-terminal — V2.4.0

**HBA, SMC**
- Spezifikation Teil 1-3 — V2.x.x

## Test/Betrieb

**Dezentrale Musterumgebung**
- Betriebskonzept — V1.1.0

**Infrastrukturdienste**
- Zulassungskonzept — V1.0.0
- Prüfvorschriften — V1.0.0
- Testspezifikation — V1.0.0

**Konnektor 1.0.0**
- Zulassungskonzept — V1.0.0
- Prüfvorschriftvorschriften — V1.0.0
- Testspezifikation — V1.0.0

**eHealth Kartenterminal 1.1.0**
- Zulassungskonzept — V1.0.0
- Prüfvorschriften — V1.0.0
- Testspezifikation — V1.0.0

**HBA, SMC**
- Zulassungskonzept — V1.0.0
- Prüfvorschriften — V1.0.0
- Testspezifikation — V1.0.0

**eGK V1.1.0/1.2.1/1.3.0**
- Zulassungskonzept — V1.0.0
- Prüfvorschriften — V1.0.0
- Testspezifikation — V1.0.0

**Musterkarten, Testlabor- und Testkarten**
- Spezifikation Musterkarten und Testkarten — V2.5.0
- Spezifikation für Testlaborkarten — V2.0.0

# Merke: Kooperation ist unumgänglich

Freie Universität Berlin

- Ein größeres Softwareprojekt ist stets so komplex, dass keine einzelne Person noch ein komplettes Verständnis davon entwickeln kann

➔ Viele Gruppen von Beteiligten

➔ Arbeit im Team (sogar in vielen parallelen Teams) ist unvermeidlich

➔ Schriftliche Dokumentation ist unvermeidlich

# Beteiligte am eGK-Projekt

- Gesetzgeber, Regierung
- Heilberufler(verbände)
  - Ärzteschaft (niedergelassen, Krankenhaus)
  - Apothekerschaft
  - Sonstige Heilberufler
- Kostenträger
  - Krankenkassen
  - Private Krankenversicherungen
  - Berufsgenossenschaften
- Patienten
  - ohne/mit Technikscheu
- Primärsystem-Hersteller
- gematik Gmbh

  AOK-Bundesverband, Arbeiter-Ersatzk.-Verband, Bundesärztekammer, Bundesknappschaft, Bundesverband der Betriebskrankenk., Bundesverband → der landwirtschaftlichen Krankenkassen, Bundeszahnärztekammer, Deutsche Krankenhausgesellschaft, Deutscher Apothekerverband, IKK-Bundesverband, Kassenärztliche Bundesvereinigung, Kassenzahnärztliche Bundesvereinigung, See-Krankenkasse, Verband der Angestellten Krankenkassen, Verband der privaten Krankenversicherung
- viele versch. Auftragnehmer
  - Siehe "Ausschreibungen" bei gematik.de (26.03.2008):

  - Aufbau/Betrieb Brokerdienst
  - System-/Service-Mgmt-Werkzeug
  - Aufbau/Betrieb Zeit- u. Namensdienst
  - Aufbau/Betrieb Zugangsnetz
  - Aufbau/Betrieb SDS
  - Aufbau/Betrieb Audit-Service
  - Aufbau/Betrieb VPN-Netze
  - Evaluation 10.000er Test usw. usf.
- HW-Lieferanten (mit viel SW)

  Chipkarten, Kartenleser, Konnektoren

# Wer sind die Nutznießer der eGK?

Freie Universität Berlin

eVerordnung:

- Ärzte:
  - Kostenaufwand f. Technik
  - Umständl. Handhabung
- Apotheker:
  - Kostenaufwand f. Technik
  + Beschleun. Erfassg./Abrechng.
- Patienten:
  - Rezepte weniger anschaulich
- Krankenversicherer
  + Beschleun. Erfassg./Abrechng.
- IT-Branche
  + bekommt viele Aufträge
- Teile der Öffentlichkeit:
  - Befürchtungen "gläserner Patient" (technisch/gesetzlich kaum begründet)

später (ePatientenakte etc.):

- Ärzte:
  + Vermeidung v. Doppelunters.
  + Vermeidung falscher Beh.
  - Vermeidung v. Doppelunters.
- Patienten:
  + bessere mediz. Versorgung
  + Kostenersparnis global

Folge all dieser Verhältnisse:

- Streit und Verzögerungen

# Zeitlicher Ablauf des Projekts

- **2002-05** Einigung Regierung/Gesundsheitswesen auf gemeinsames Vorgehen
- **2003-03** Start Projektgruppe Gesundheitsminist.
- **2003-08** Projektvergabe Erstellg. Rahmenarchitekt. (RA)
- **2003-10** Verabschiedung GKV-Modernisierungsgesetz (GMG).
  - z.B. §291a(1) SGB V: "Die Krankenversichertenkarte [...] wird **bis spätestens zum 1. Januar 2006** [...] zu einer elektronischen Gesundheitskarte erweitert.
- Start der Umsetzung
- **2005-01** gematik gGmbh für Umsetzung gegründet
- **2005-03** Konzeption (RA+LA) fertig (26 Dok., 2000 Seiten)
- Start d. techn. Implementierg.
- **2006-12** erste Feldtests; nur Versichertenstammdaten lesen
- **2007-06** (geplant: 2005-12) erste 10.000er Feldtests mit eRezept
- **2008-09** erstes Kartenterminal zertifiziert

# Zeitlicher Ablauf des Projekts (2)

- **Erinnerung: 2003-10:** §291a(1) SGB V: "Die Krankenversichertenkarte [...] wird **bis spätestens zum 1. Januar 2006** [...] zu einer elektronischen Gesundheitskarte erweitert.
- **2007-2019: Streit um**
  - Kosten f. Ärzte
  - "Gläserner Patient"
  - Praktikabilität einer PIN (insbes. für Ältere)
  - Praktikabilität u. Bedien-aufwand beim eRezept (Ärzte)
  - Authentisierung des Fotos
  - Stammdatenaktualisierung in der Arztpraxis
  - eArztbrief
  - 2. Generation d. Chipkarte
- **Stand 2020-04**: die eGK liefert nur Stammdaten (Name, Adresse, Geburtstag, Versich.)
  - Wie früher die KVK, nur ohne Längenbeschränkung
  - eRezept aktuell geplant für 2021, mit Smartphone-App

Warum ist das so furchtbar in die Hose gegangen?

# Merke: Großprojekte sind riskant und geraten meistens in Zeitverzug

Freie Universität Berlin

Eine der wichtigsten Erkenntnisse der Softwaretechnik:

- Je größer ein Projekt, desto riskanter ist es
  - Durch technische Komplexität der Realisierung
  - Durch komplexe, schlecht verstandene Anforderungen
  - Durch im Zeitverlauf veränderliche Anforderungen
  - Durch viele Beteiligte mit unterschiedlichen Interessen

Eines der wichtigsten Ziele der Softwaretechnik ist deshalb:

- Projekte in verdauliche Happen aufteilen
  - Gleichzeitige Elemente, die gut von einander isoliert sind
    - Module, Subsysteme, Dienste ➔ Entwurf
  - Schrittweise Entwicklung in mehreren Schüben
    - Iterationen, idealerweise agil ➔ SW-Prozess

# Danke!

Es folgen:
Bonusfolien über digitale Signatur

# Exkurs: Digitale Signatur

Inhalt des Exkurses (je 1 Folie):

- Verschlüsselung
- Symmetrische Verschlüsselung
- Asymmetrische Verschlüsselung
- Digitale Unterschrift (digitale Signatur)
- Public-Key-Infrastruktur
- Ermöglichte Sicherheitseigenschaften

- Dieser Exkurs deckt nur ab, was jede/r Softwareingenieur/in wissen sollte
  - unzählige Details fehlen

# Verschlüsselung

- Sei gegeben eine Nachricht **K** ("Klartext"), die geheim gehalten werden soll
- und zwei Funktionen
  - **V** ("Verschlüsselung") mit V(K) = **C** ("Chiffretext")
  - **E** ("Entschlüsselung") mit E(C) = K
- so, dass C jedem unbefugten Empfänger ("Angreifer") unverständlich ist
  - C sieht aus wie eine Folge von Zufallsbits: total redundanzfrei
- dann nennt man (V,E) ein Verschlüsselungsverfahren

- Offensichtlich müssen sich Sender und Empfänger auf das Paar (V,E) einigen und E geheim halten
  - Solche Verfahren wurden bereits in der Antike erfunden

# Symmetrische Verschlüsselung

Modernerer Ansatz:

- V und E sind beide allgemein bekannt
  - viele Personen haben sie als Software auf ihrem Computer
- aber sie sind parametrisiert mit einer großen Zahl S
  - $V_S$, $E_S$
  - $E_S(V_S(K)) = K$, aber es gibt viele mögliche S (z.B. $2^{128}$ Stück), die ein Angreifer durchprobieren muss
    - Wie lange dauert das? → Überschlagsrechnung
- Sender und Empfänger müssen sich nun nur noch auf S einigen
  - → ermöglicht den häufigeren Austausch der Verschlüsselung
    - Verfahren heißt *symmetrisch*, weil Sender und Empfänger denselben Schlüssel S benutzen
  - **Aber wie vereinbart man einen solchen Schlüssel, ohne belauscht zu werden?**

# Asymmetrische Verschlüsselung

Noch modernerer Ansatz:

- V und E sind beide allgemein bekannt
  - viele Personen haben sie als Software auf ihrem Computer
- aber sie sind parametrisiert mit zwei verschiedenen großen Zahlen P und Ö, die zusammengehören
  - $E_P(V_Ö(K)) = K$
  - P heißt der private Schlüssel und ist geheim
    - Er wird nie (nie! nie!!) weitergegeben
  - Ö heißt der öffentliche Schlüssel und wird quasi im Telefonbuch bekannt gegeben → Schlüsselvereinbarung entfällt!
  - Jeder Benutzer hat sein eigenes Paar (P, Ö)
- Somit kann jeder dem Empfänger eine Nachricht senden, die nur dieser entschlüsseln kann
  - Verfahren heißt asymmetrisch, weil Sender und Empfänger verschiedene Schlüssel benutzen

# Digitale Unterschrift (digitale Signatur)

Freie Universität Berlin

Jetzt kommt ein toller Kniff:

- Asymmetrische Verschlüsselung funktioniert auch "andersrum":
  - $E_{Ö}(V_P(K)) = K$
  - Dann kann also jeder mit Ö die Nachricht entschlüsseln
- Wozu ist das gut?
  - Ich sende Nachrichten der Form "Von prechelt@inf.fu-berlin.de: $V_P(K)$", wobei $P = P_{Prechelt}$
  - jetzt kann jeder Empfänger nachprüfen, dass die Nachricht wirklich von Prechelt kommt
  - denn dann (und nur dann) kann man sie mit $Ö_{Prechelt}$ entschlüsseln
  - Kein Angreifer kann die Nachricht gezielt verfälschen.
- Die genialste Mathe-Erfindung der letzten 100 Jahre

# Public-Key-Infrastruktur (PKI)

Zwei Probleme bleiben übrig:

- Das ganze bricht zusammen, falls
  (a) ein Angreifer A das Telefonbuch fälschen kann
  - und mir sein eigenes $Ö_A$ als $Ö_{Prechelt}$ unterjubelt, oder aber
- (b) jemand meinen privaten Schlüssel ausspioniert
- (a) Deshalb müssen Telefonbucheinträge wiederum unterschrieben sein
  - von jemand, dem alle vertrauen und dessen Ö quasi jeden Tag in der Tagesschau etc. durchgesagt wird, so dass es niemand verfälschen kann.
  - → *Zertifizierungsstellen* (Certification Authorities, CAs)
  - Der ganze technisch-organisatorische Rahmen samt Verfahren heißt *Public-Key-Infrastruktur (PKI)*
- (b) Schlüssel P darf niemals vertrauenswürdige Hardware verlassen → Chipkarte führt Signatur durch

# Ermöglichte Sicherheitseigenschaften

Verschlüsselung:

- Vertraulichkeit (privacy)
  - Angreifer kann verschlüsselte Nachricht nicht unbefugt lesen
- Integrität (integrity)
  - Angreifer kann verschlüsselte Nachricht nicht gezielt ändern

Signatur:

- Authentizität (authenticity)
  - Empfänger kann Echtheit von Absender/Signatur überprüfen
- Nicht-Abstreitbarkeit (non-repudiation)
  - Sender kann nicht abstreiten, eine unterschriebene Nachricht gesendet zu haben

- Man kann Signatur und Verschlüsselung auch zugleich anwenden und somit alle 4 Eigenschaften zugleich erreichen

# (Ende des Exkurses über digitale Signatur)

Noch zwei Anmerkungen:

- Zur Verschlüsselung:
  - Asymmetrische Verschlüsselung ist sehr rechenaufwändig,
  - deshalb verschlüsselt man in der Praxis größere Datenmengen stets symmetrisch
  - und benutzt asymmetrische Verfahren nur für die dazu nötige Schlüsselvereinbarung.

- Zur Signatur:
  - Zur digitalen Signatur verwendet man in der Praxis ebenfalls nicht das ganze Dokument
  - sondern signiert nur einen Hashwert (z.B. 256 bit lang).
  - Dafür gibt es spezielle *kryptografische Hashfunktionen*