Willkommen zur Vorlesung

Softwarearchitekturen im Finanz- und Versicherungsbereich

im Sommersemester 2010

Prof. Dr. Jan Jürjens

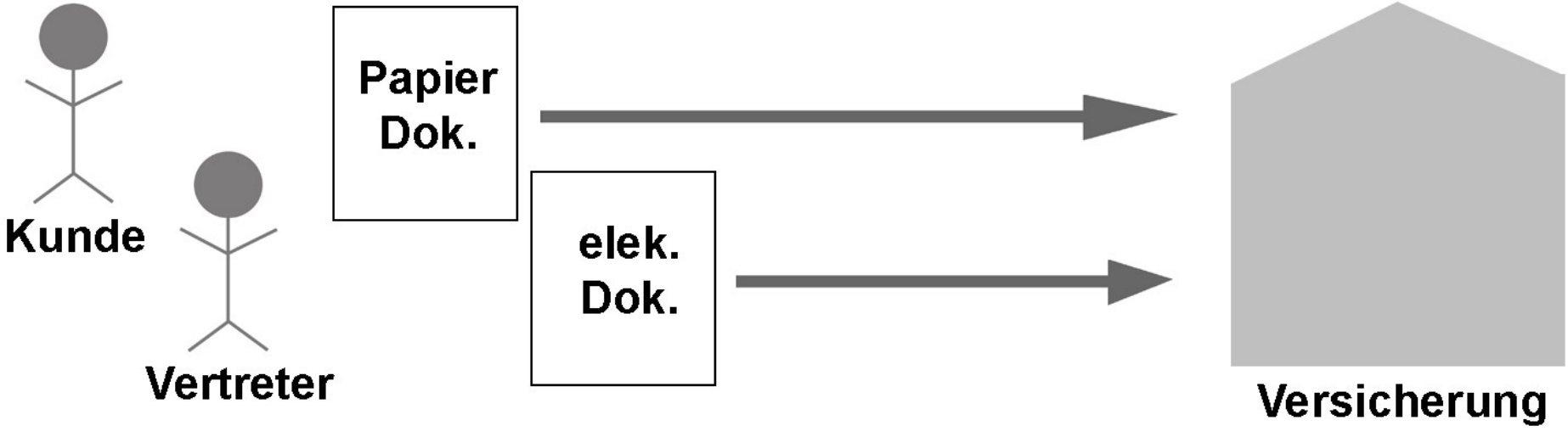
TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

9. Software Architektur Anwendungsbeispiele

Sicherer Einsatz elektronischer Unterschriften im Versicherungswesen

Motivation

- 2 Versionen des Antrags wegen Unterschrift
- Zusammenführung notwendig
- Medienbruch verursacht Kosten



Lösungsmöglichkeit

- Kunde unterschreibt auf Pad
- Unterschrift ins digitale Dokument einbinden
- Archivierung und Weiterverarbeitung des elektronischen Antrags

Bedrohungen

- für den Kunden
  - seine Unterschrift wird missbraucht
  - sein Antrag wird verändert
- für die Gesellschaft
  - Kunde kann Antrag anzweifeln
  - Vertreter kann Antrag manipulieren

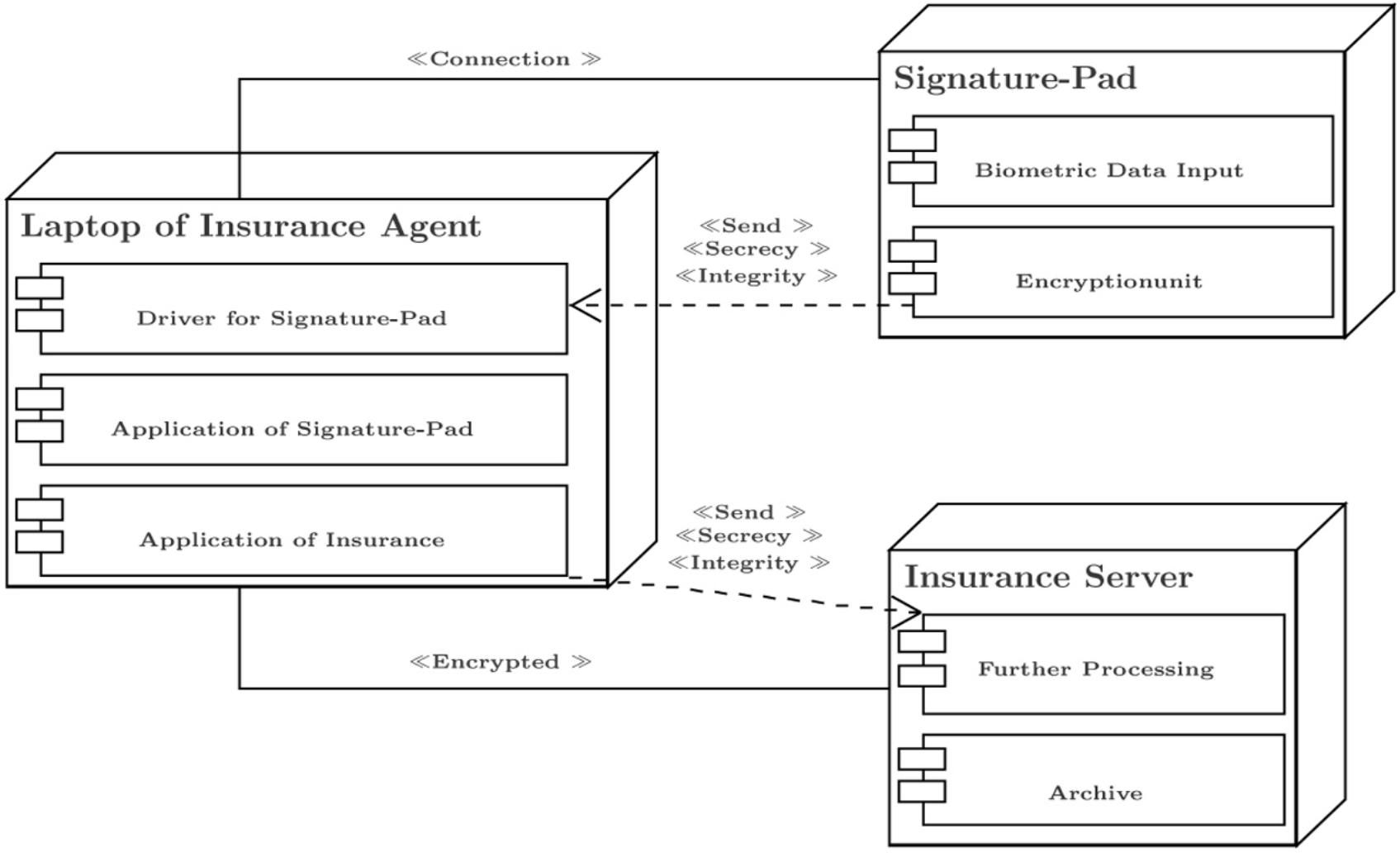
Sicherheitsziele

- Vertraulichkeit der biometrischen Daten
- Verbindlichkeit der Kunden-Unterschrift
- Integrität des Dokuments

Geräte zur Unterschriftenerfassung

- Statische Informationen
  - Bild
- Dynamische Informationen
  - Druck
  - Zeit

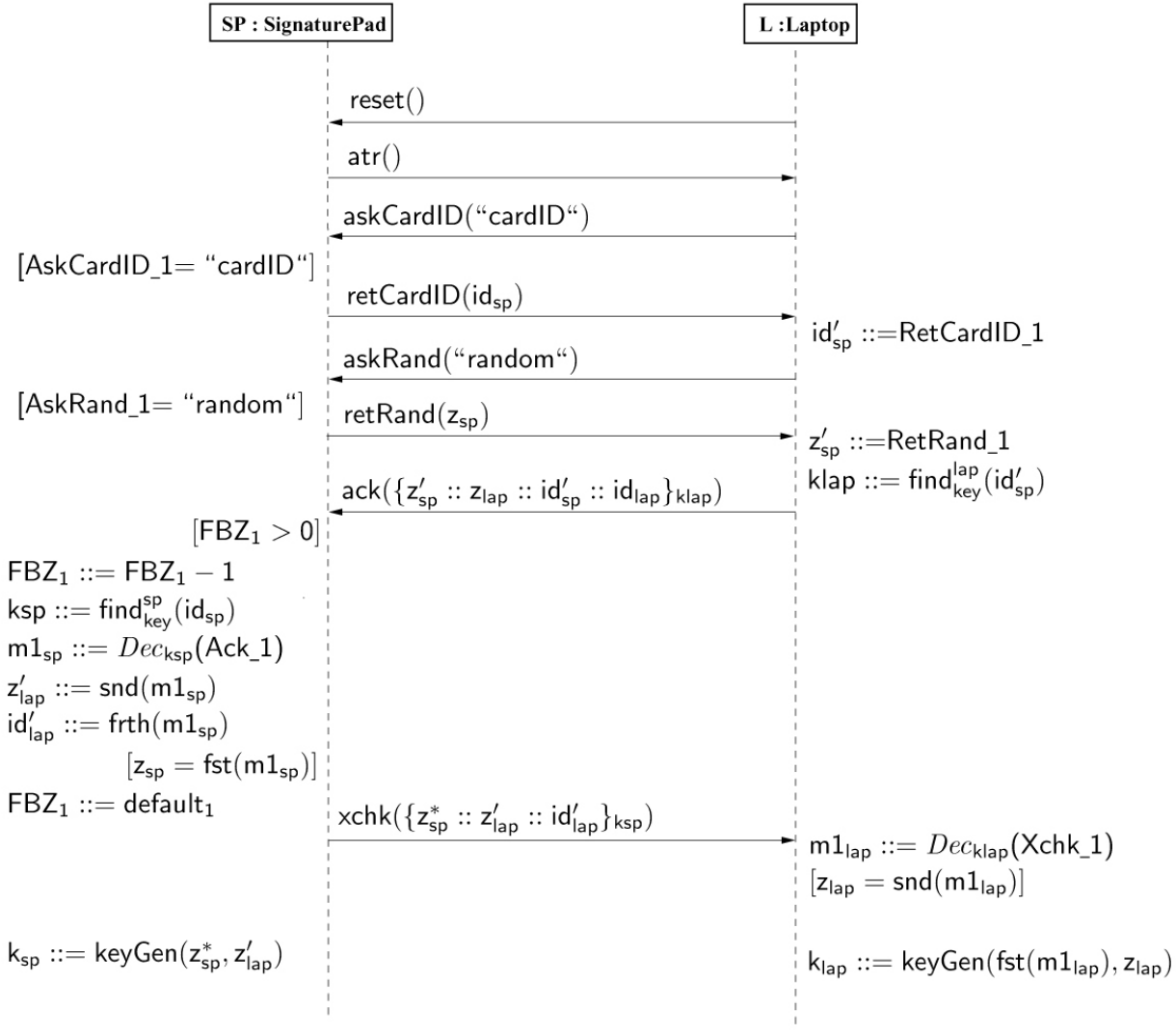
Kommunikationspartner



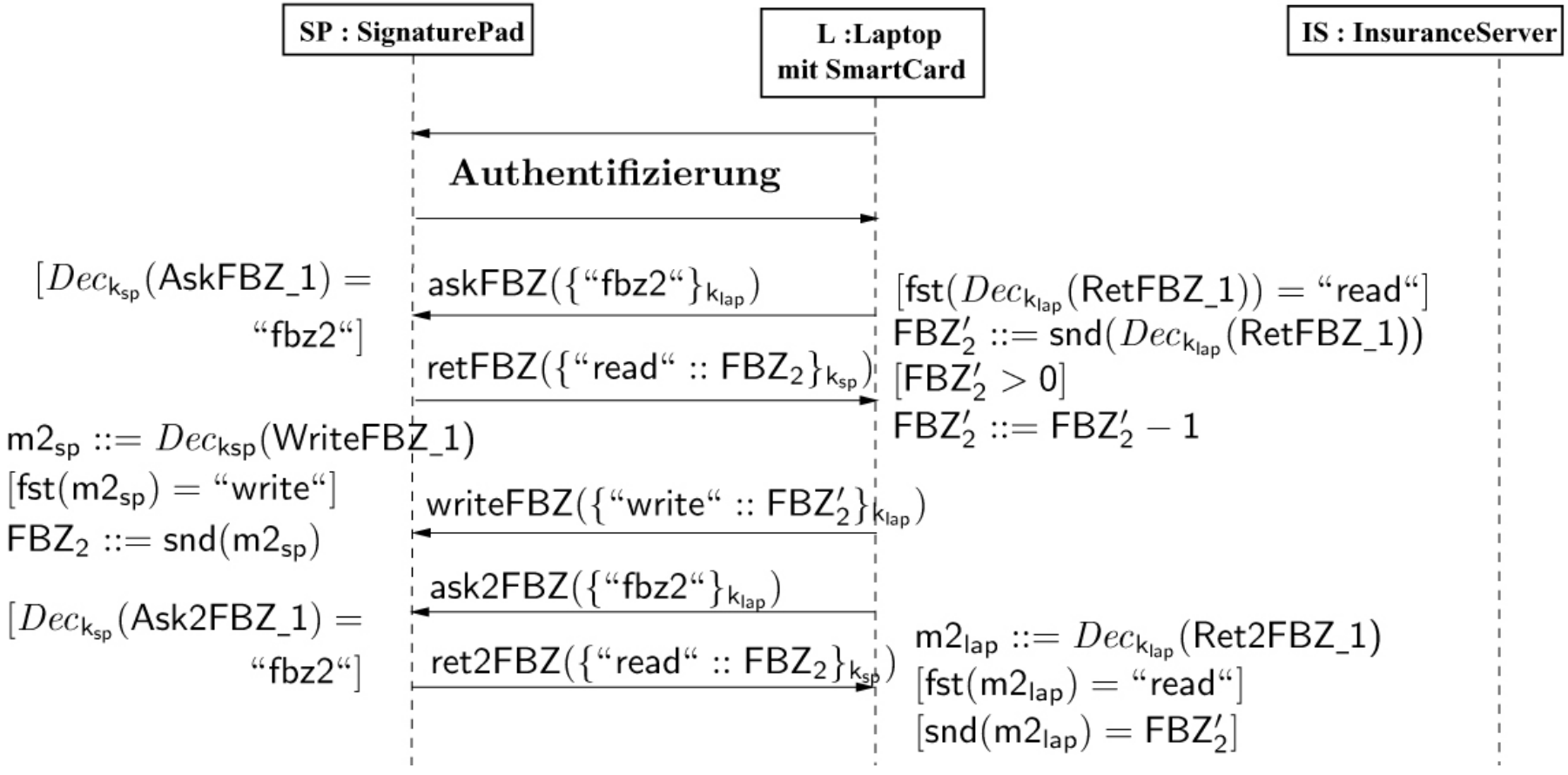
Protokoll mit SmartCard

- Orientierung am biom. Zugangsverfahren
- Authentifizierung zwischen Laptop und Pad
- SmartCard im Laptop des Vertreters
- Kontrolle der Signatur auf der SmartCard
- Anforderung der Referenzdaten zum Vergleich von der Gesellschaft

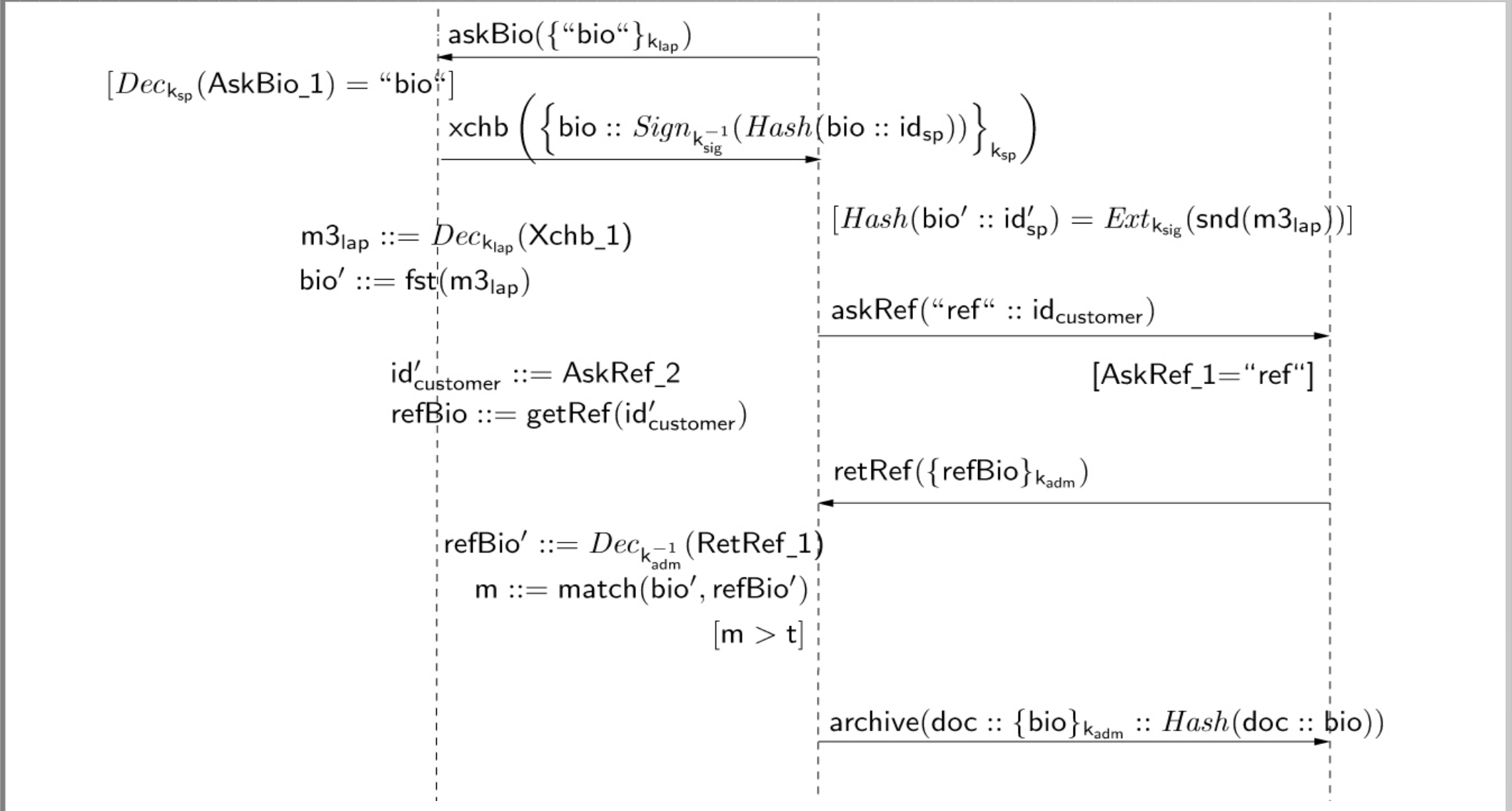
Authentifizierung 1



Authentifizierung 2



Authentifizierung 3



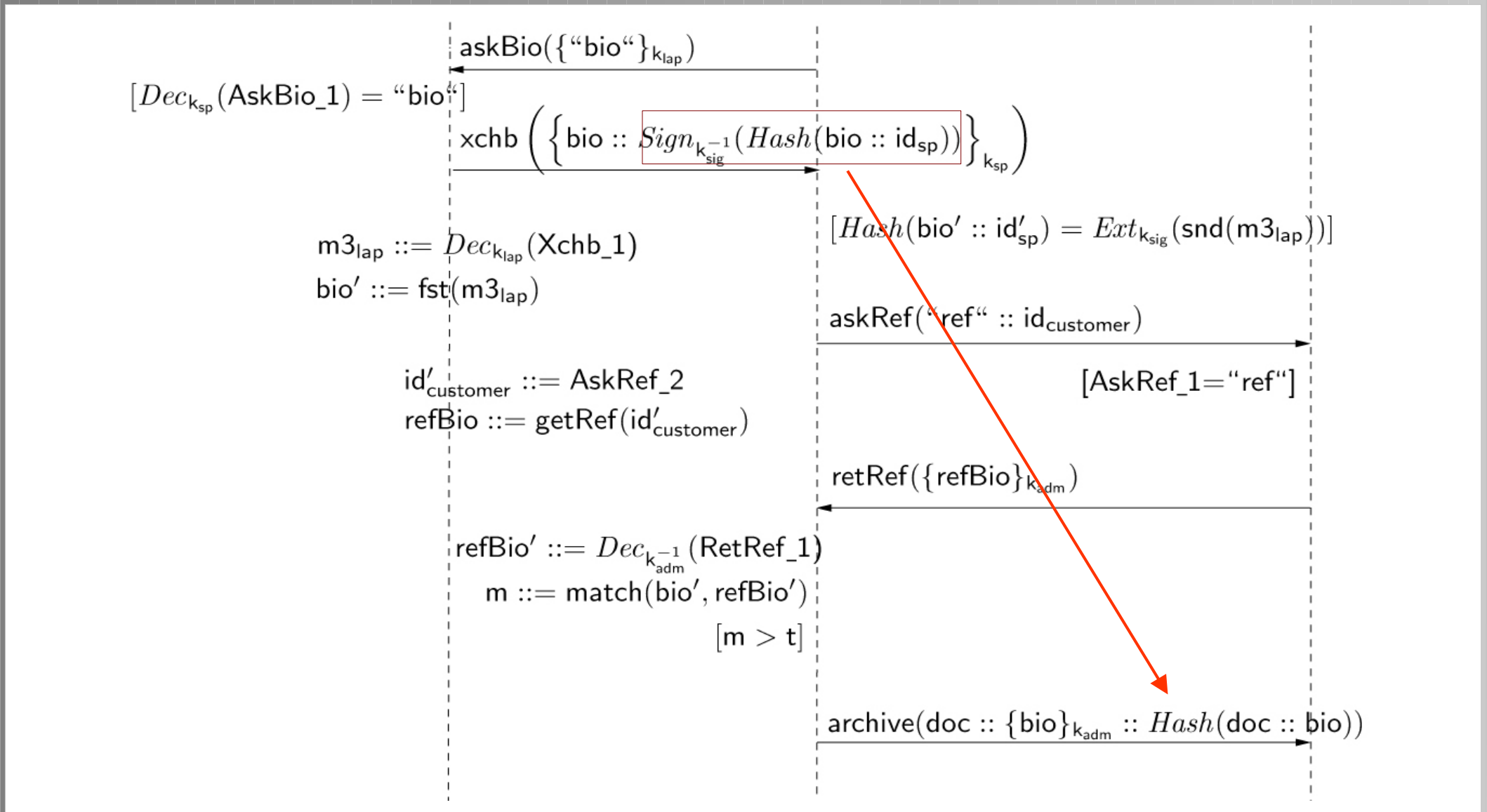
Sicherheitsanalyse

- Keine Möglichkeiten für Standardangreifer
- Verbindlichkeitsprüfung durch

```
Input_formula(attack, conjecture, (
   ( (knows(bio) | knows(refbio) )
    & knows(doc)
    & knows(k_adm) )
)).
```

- kennt Angreifer biom. Daten, dann ist eine Attacke möglich

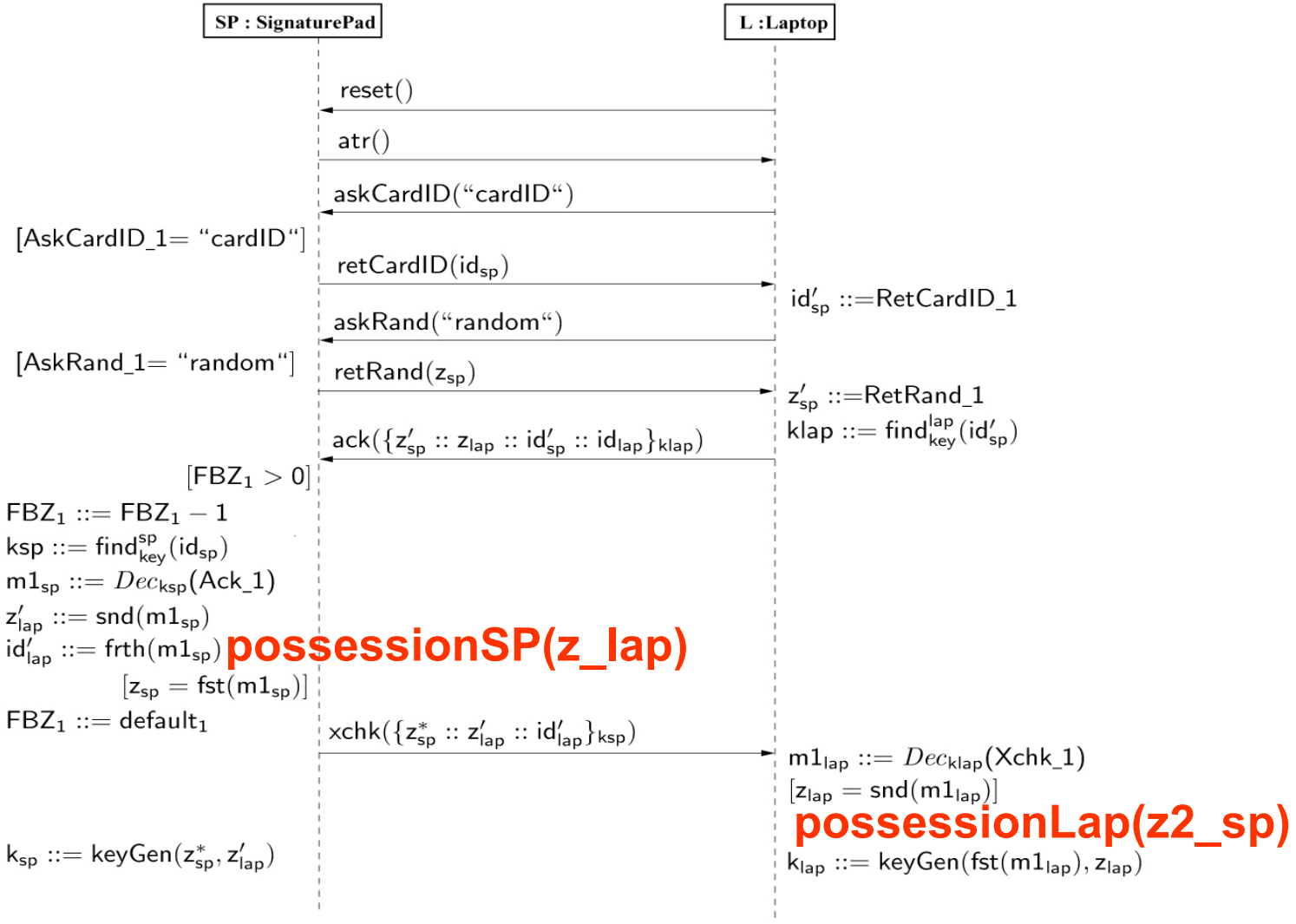
Sicherheitsanalyse



Authentizitätsprüfung

- ist erfolgreich, falls Sitzungsschlüssel korrekt berechnet wird
- notwendig dazu
  - Zufallszahlen von Laptop und Pad
  - initialer sym. Schlüssel
- Überprüfung mit possession-Prädikat

Authentizitätsprüfung



Setzen der Prädikate

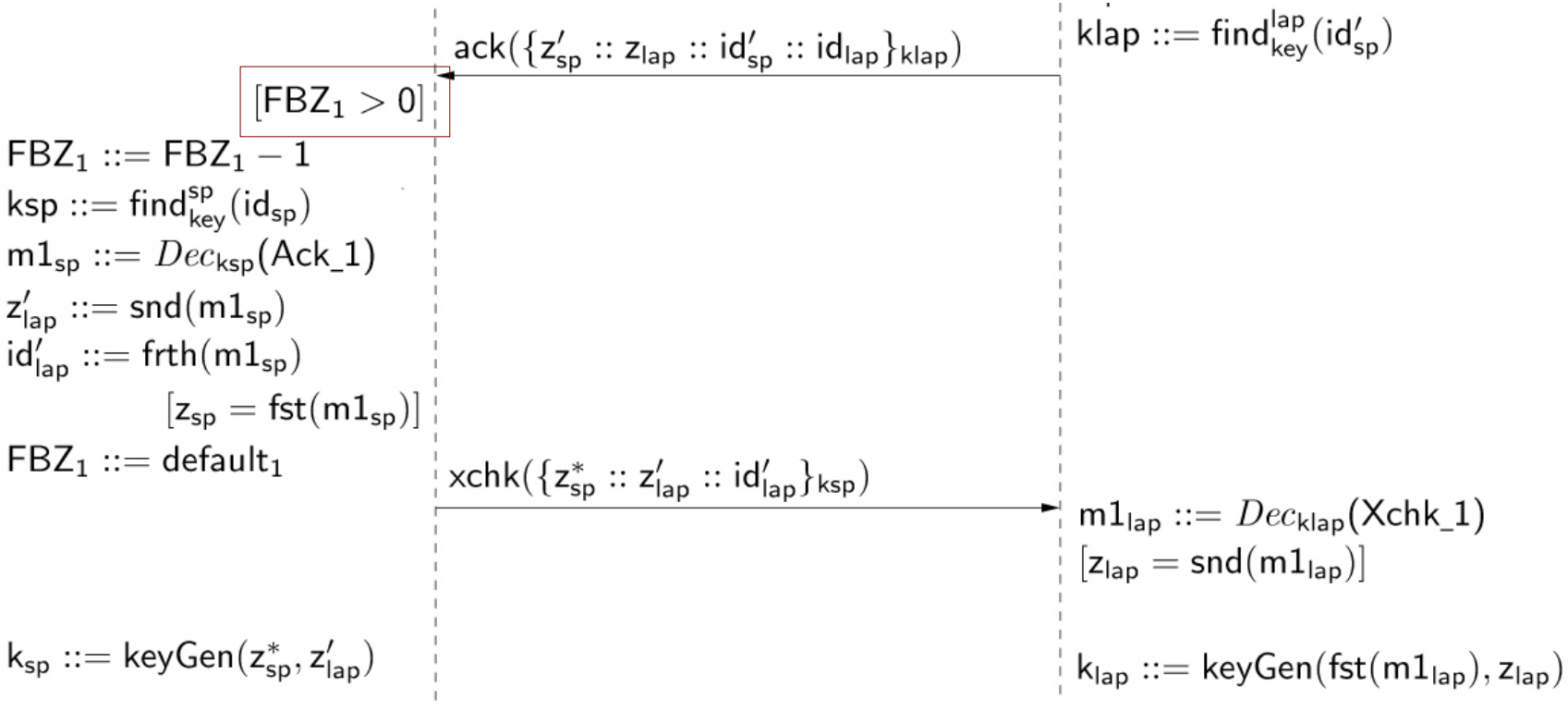
- Komponenten kennen eigene Zufallszahlen
- Authentifizierungsschlüssel beiden bekannt

```
Input_formula(attack, conjecture, (
   ( possessionLap(findKey)
   & possessionLap(z2_sp)
   & possessionLap(z_lap) )
)).
```

- Prüfung erfolgreich

Gegenbeispiel

- Erster Fehlbedienungszähler abgelaufen
- als Fakt: `~(greater(fbz1,zero))`



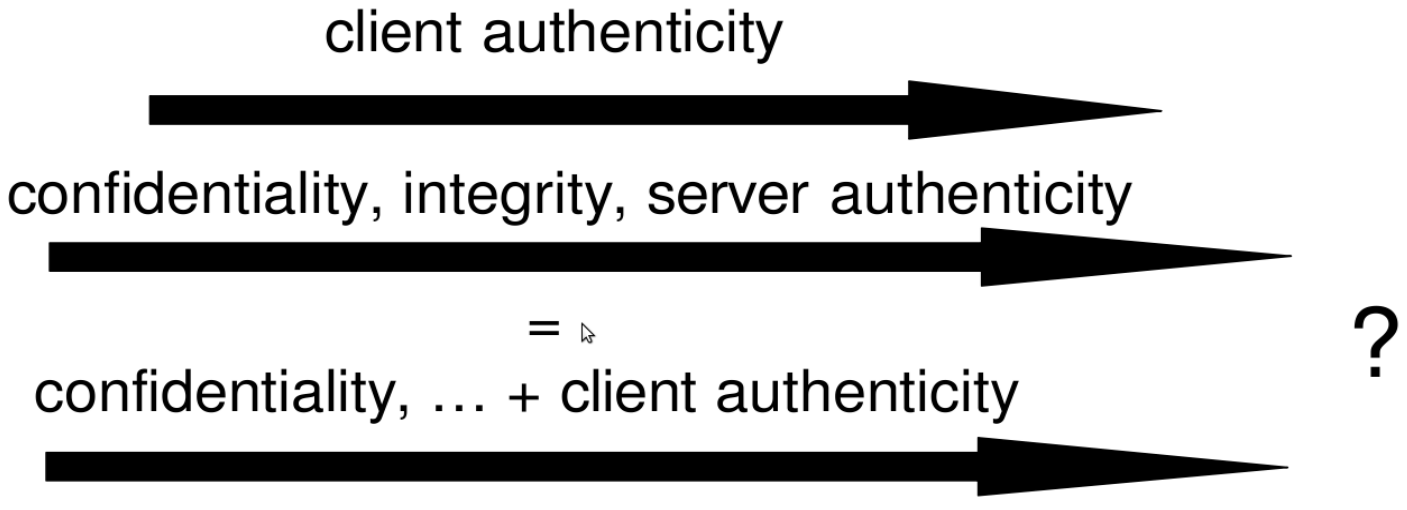
Fazit

- Elekt. Unterschrift flexibel einsetzbar
  - für firmeninterne Dokumente
  - für Angebote von Zulieferern
- Protokoll notwendig, um Verbindlichkeit zu sichern
- bisher kein Gerichtsurteil über den Einsatz elektronischer Unterschriften

Modelling and Verification of Layered Security Protocols: A Bank Application

Layered Security Protocols

- Protokoll einer höheren Schicht nutzt Dienste eines Protokolls das auf einer niedrigeren Schicht angesiedelt ist
- Die große Frage ist dann: Addieren sich Sicherheitseigenschaften auf?
- Wünschenswert: Secure Channels Abstraktion



Auto Fokus

- Analyse unter anderem mit Auto Fokus
  - CASE tool mit formaler Basis
  - Grafisch auf Basis einer UML ähnlichen Modellierungssprache
    - System Structure Diagrams, State Transition Diagrams, Message Sequence Diagrams, Data Types Definitions
  - Bietet
    - Simulation
    - Validation (Konsistenz, Tests, Model Checking)
    - Codegenerierung (Java, C)

Die Banken Applikation

- Sicherheitsanalyse einer web basierten Bankenapplikation einer großen deutschen Bank
- Hauptanforderungen
  - Vertraulichkeit personenbezogener Daten
  - Unabstreitbarkeit

Die Banken Applikation

- Zwei Schichten Architektur
  - Verbindungsaufbau über SSL (erste Schicht)
    - Wird für Integrität, Vertraulichkeit und Server - Authentifizierung genutzt
    - Keine Client – Authentifizierung
  - Client Authentifizierung über selbst entwickeltes Protokoll (zweite Schicht)
    - Nutzt Sessionkey des SSL Protokolls für Verschlüsselung

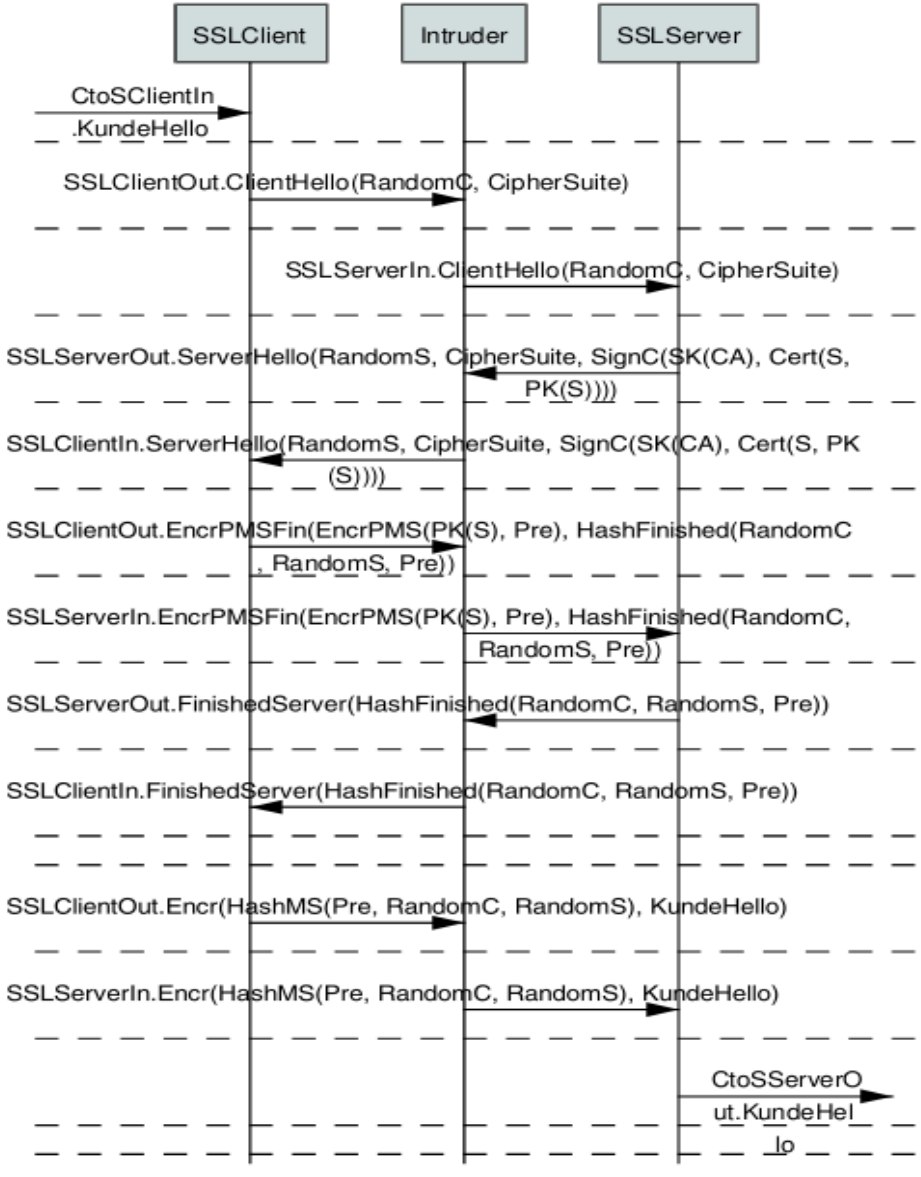
Generic Adversary (Dolev-Yao) I

- May know some data in advance (but not private keys etc.).
- Can read, modify, split, recombine, insert and delete messages.
- Not able to derive private key from public key.
- Can only read encrypted messages with the corresponding key.
- Cannot break encryption, fake signatures etc.
- Cannot perform statistical attacks.

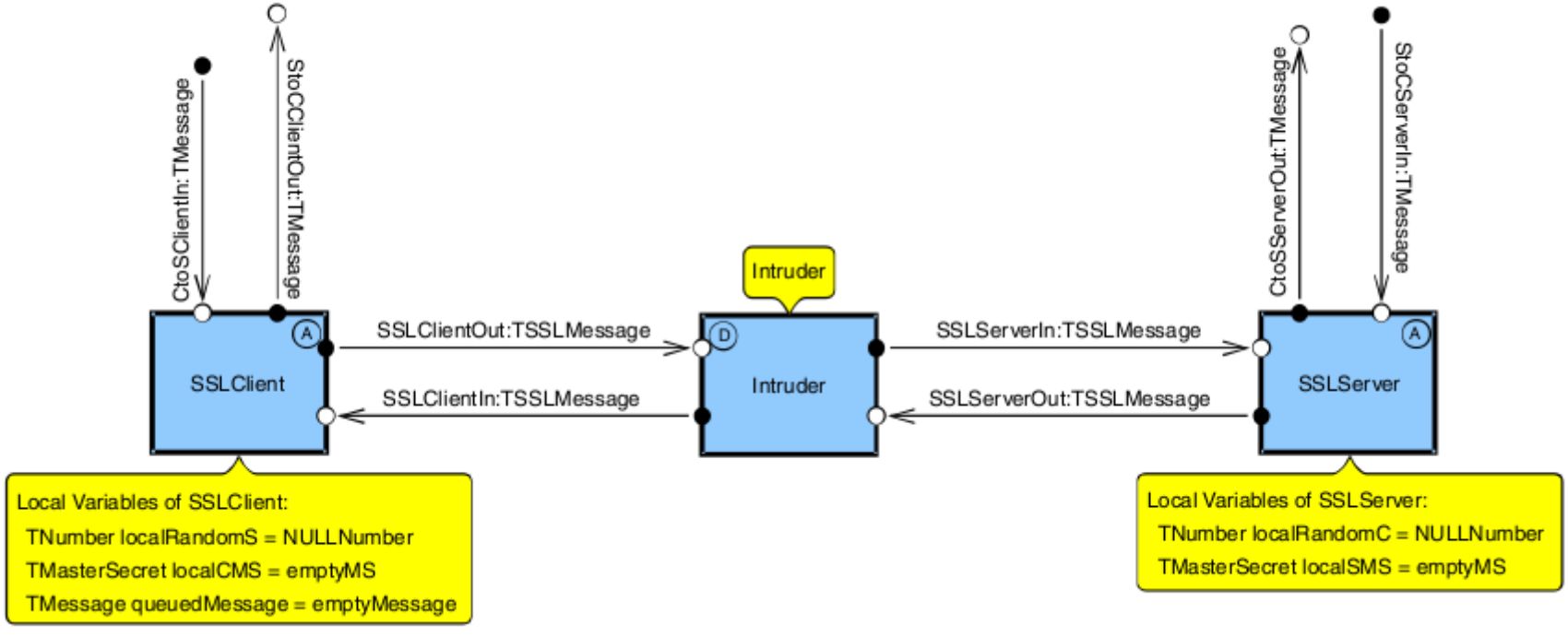
Generic Adversary (Dolev-Yao) II

- Adjust adversary model to account for SSL security properties.
- Justify that specialised adversary model wrt. top-level protocol is as powerful as generic adversary wrt. Protocol composition.
- Verify top-level protocol wrt. specialised adversary.
- Implies verification of protocol composition

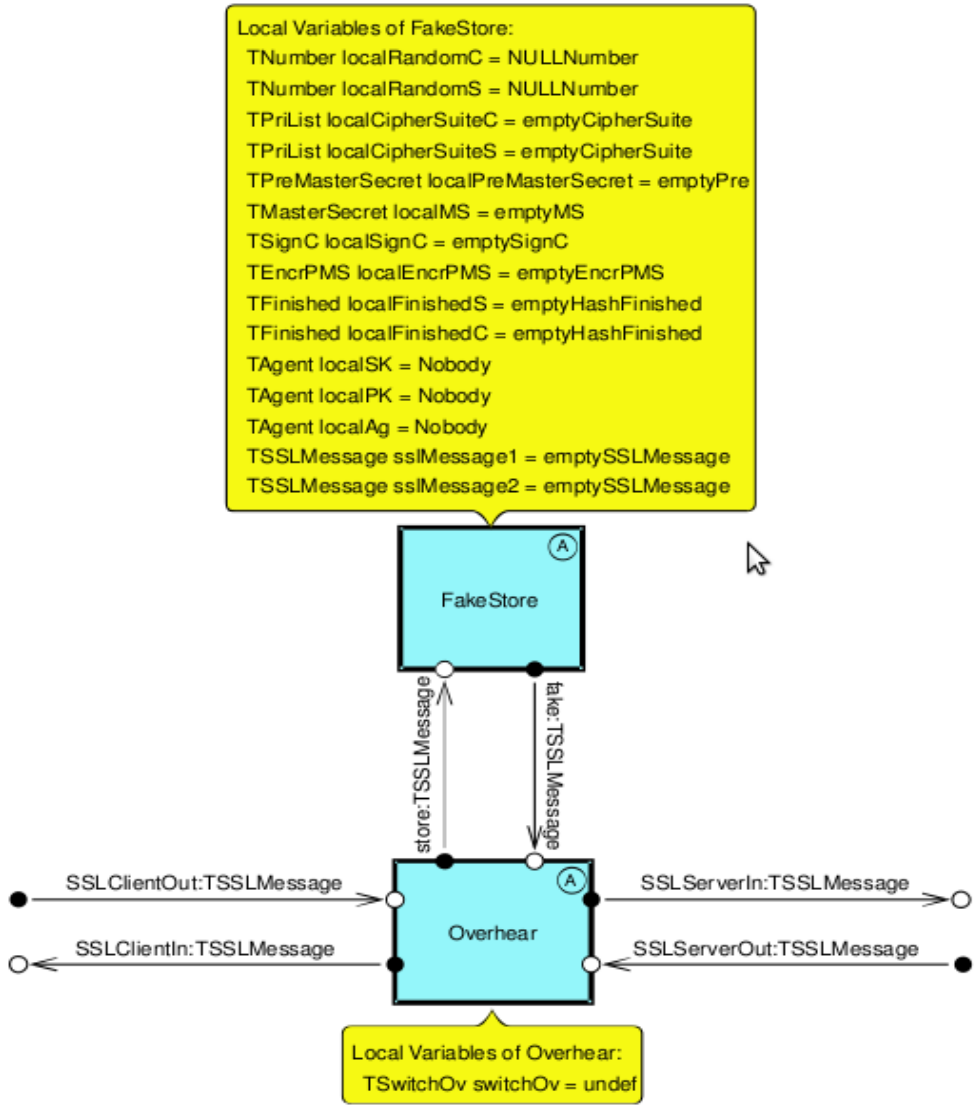
SSL Protokoll mit Angreifer



SSL Protokoll mit Angreifer Modelliert in AutoFokus



SSL Protokoll mit Angreifer
Modellierung Angreifer



Authentifikation Webserver

- Angreifer darf sich nicht erfolgreich authentifizieren gegenüber dem Client
- Das wäre der Fall wenn der Client eine GotServerFinished Msg akzeptiert bevor der Webserver im Zustand GotPreMasterSecret ist
- Ausgedrückt in CTL Logik:
  - ¬(E(¬Server im Zustand »GotPreMasterSecret« U (Client im Zustand »GotServerFinished«)))

Authentifikation Webserver

- Ergebnis durch den AutoFokus Validator:
  - Angreifer kann diese Zustandskombination nicht erzeugen
- Authentifikation des Webservers gegenüber dem Client gesichert

Vertraulichkeit SSL

- Angreifer darf nicht der Lage sein das Mastersecret aus den Handshake Nachrichten abzuleiten
  - Wenn er dazu in der Lage wäre könnte er die nachfolgenden Nachrichten entschlüsseln
- Ausgedrückt in CTL Logik:
  - AG((SSLClient im letzten Zustand ∧ SSLServer im letzten Zustand) ⇒ ((MasterSecret FakeStore = MasterSecret SSLClient) ∧(MasterSecret FakeStore = MasterSecret SSLServer)))

Vertraulichkeit SSL

- Ergebnis des AutoFokus Model Checkers:
  - Angreifer kann das Mastersecret nicht ableiten.
- Anforderungen an das Protokoll zur Absicherung der ersten Schicht also erfüllt.

Client Authentifizierungsprotokoll

- 1. Der Kunde schickt eine KundeHello-Nachricht, um anzuzeigen, dass er sich mit dem Webserver verbinden möchte. Auf diese Nachricht hin wird eine SSL-Verbindung aufgebaut.
- 2. Nachdem eine SSL-Sitzung etabliert wurde, generiert der Webserver eine Zufallszahl und schickt sie an den Kunden.

Client Authentifizierungsprotokoll

- 3. Der Kunde signiert die Zufallszahl mit seinem privaten Schlüssel und sendet sie zusammen mit seinem Zertifikat an den Webserver zurück. Die Signatur und das Zertifikat werden vom Webserver geprüft und die signierte Zufallszahl wird mit der vorher gesendeten verglichen. DerWebserver nimmt eine Plausibilitätsprüfung der GlobalID vor und speichert diese, da sie ihm vorher nicht bekannt ist. Bei erfolgreicher Ausführung der Prüfungen ist der Authentifizierungsvorgang abgeschlossen.

Client Authentifizierungsprotokoll

- 4. Der Webserver sendet ein leeres Formular zusammen mit der GlobalID an das Backendsystem. Dort wird es mit den gespeicherten Kundendaten befüllt.
- 5. Das mit den Kundendaten befüllte Formular wird an den Kunden zurückgeschickt.

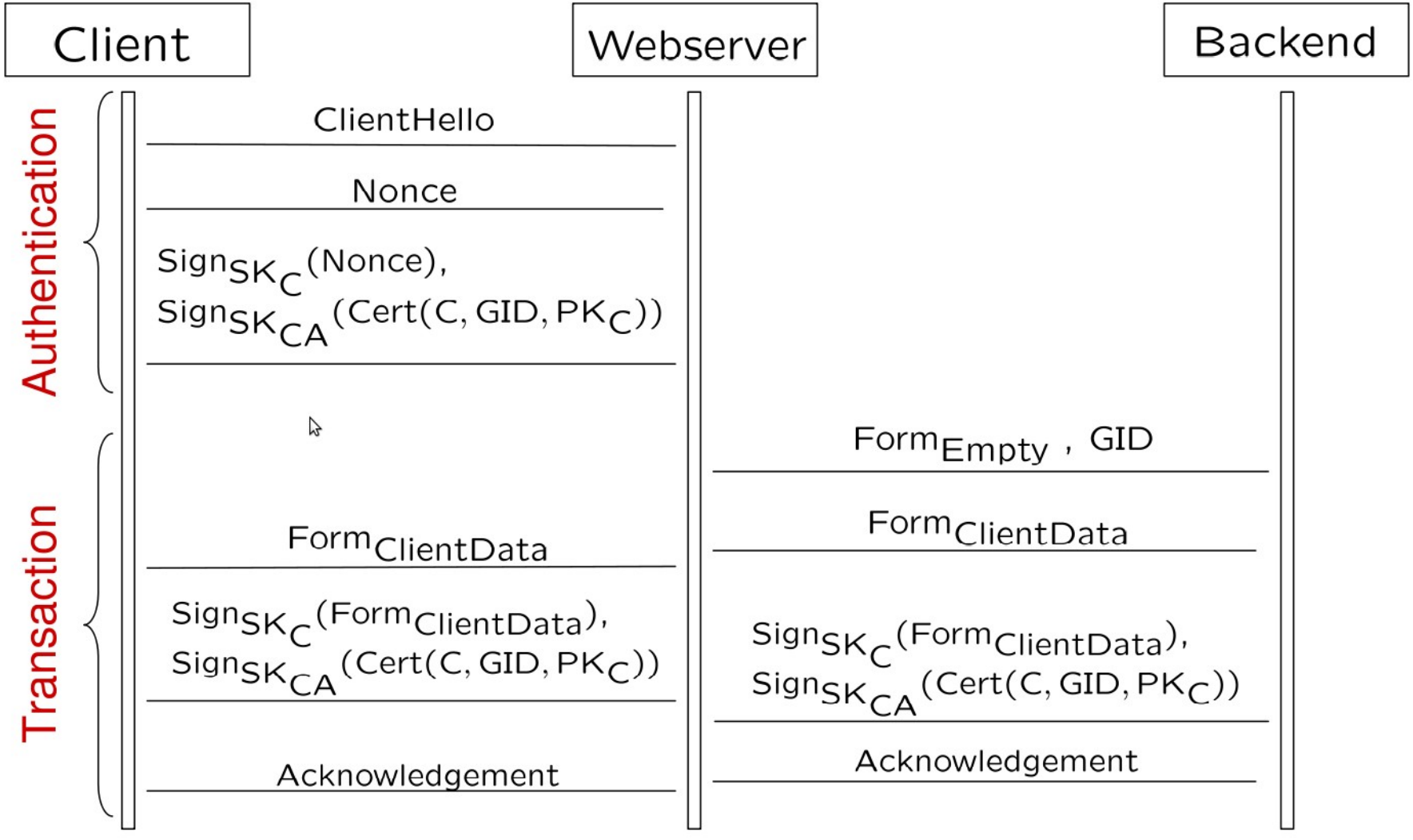
Client Authentifizierungsprotokoll

- 6. Der Kunde signiert die Kundendaten mit seinem privaten Schlüssel. Die Signatur seiner Daten dient als elektronische Unterschrift, also zur Bestätigung seines Auftrags. Im Anschluss sendet er die signierten Kundendaten an das Backendsystem zurück. Das Backendsystem prüft das Zertifikat und die Signatur, sowie die im Zertifikat enthaltene GlobalID mit der zuvor gespeicherten. Der Vergleich der empfangenen Kundendaten mit den im System gespeichertendient dazu, zu überprüfen, ob die Daten verändert wurden.

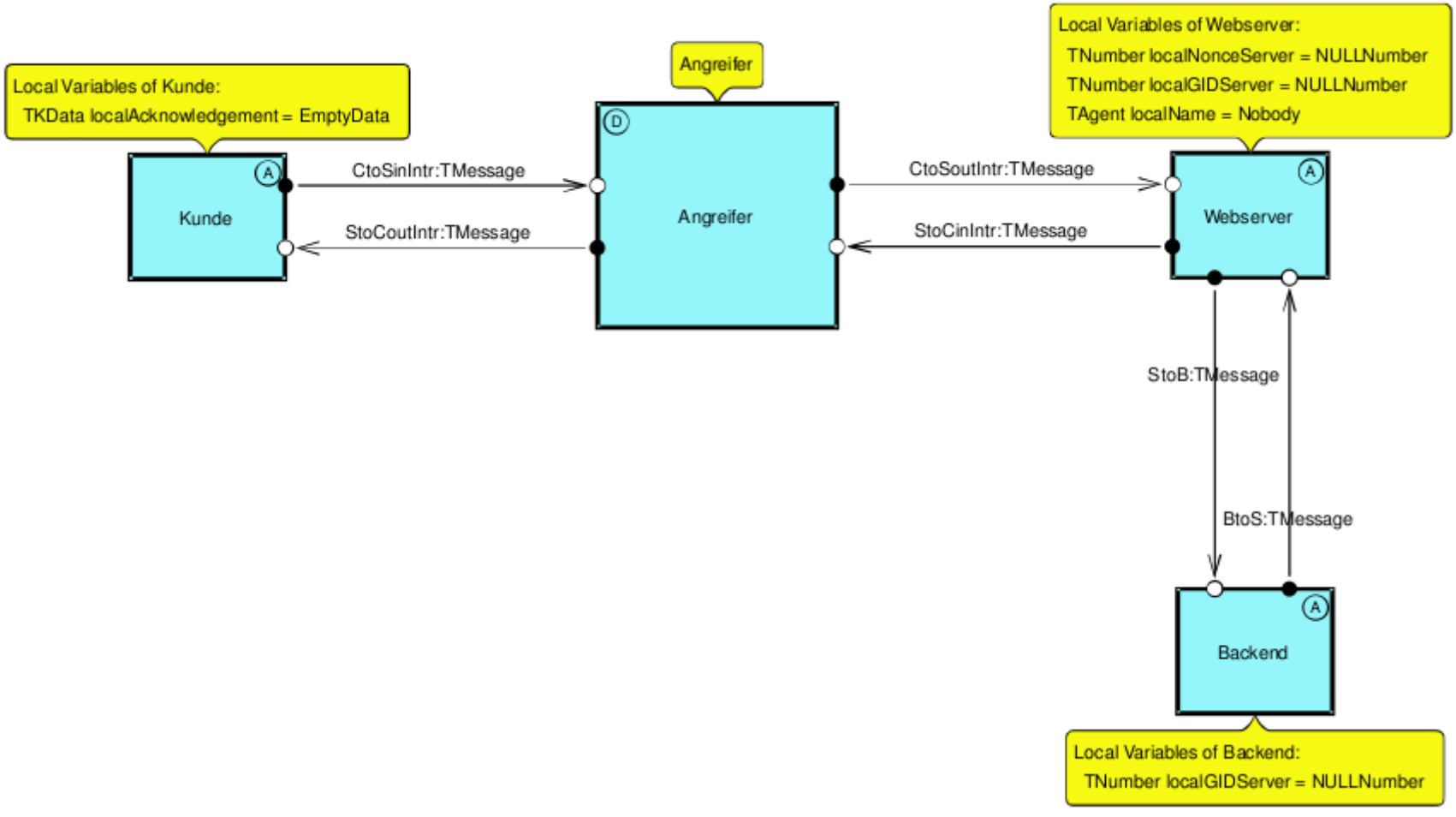
Client Authentifizierungsprotokoll

- 7. Sind die Prüfungen erfolgreich verlaufen, wird der Auftrag generiert und dem Kunden eine Auftragsbestätigung gesandt.
- 8. Das Ende-Signal kann ein Logout-Vorgang des Kunden oder ein Timeout sein. In diesen Fällen wird die Sitzung zwischen Server und Backend beendet.

Client Authentifizierungsprotokoll



Authentifizierungsprotokoll Modelliert in AutoFokus



Client Authentifikation

- Angreifer darf nicht in der Lage sein eine Nonce mit dem Zertifikat des Kunden zu signieren
  - Wenn also der Kunde keine signierte Nonce gesendet hat darf der Webserver nicht im Zustand GotSignedNonce sein und die darin enthaltene Identität C gespeichert haben
- In CTL:
  - ¬(E(¬Kunde im Zustand »SentNonceCert« U (Webserver im Zustand »GotSignedNonce« ∧Webserver hat C gespeichert)))

Client Authentifikation

- Prüfung durch Modellchecker ergibt das der Angreifer nicht in der Lage ist gegenüber dem Webserver die Identität eines dritten vorzutäuschen

Vertraulichkeit Kundendaten

- Angreifer darf zu keinem Zeitpunkt in der Lage sein die vertraulichen Kundendaten einzusehen.
  - Dazu gehört die GlobalID die unter Umständen über Drittsysteme den Zugriff auf die Kundendaten erlaubt
  - Dazu gehören alle Formulardaten die zwischen Backend und Client ausgetauscht werden
- In CTL:
  - AG ((FakeStore erhält die GlobalID nicht) ∧ (FakeStore erhält die Kundendaten nicht) ∧ (FakeStore erhält die Bestätigung für den Kunden nicht))
- Ergebnis Model Checker: Vertraulichkeit ist gewährleistet

Ergebnis

- Gesamtprotokoll wird als sicher angesehen
- Ergebnisse aus der Analyse der ersten Protokollschicht wurden beim modellieren der zweiten Protokollschicht als Annahmen für das Angreifermodell berücksichtigt.
- Aber:
  - Keine Modellierung des Gesamtprotokolls
  - Keine direkte Verifikation des Gesamtprotokolls
  - Überführung von Eigenschaften der ersten Schicht in Annahmen für die zweite Schicht nicht als korrekt bewiesen
  - Keine Untersuchungen was Verletzungen der Annahmen zur Folge hätten.

End-to-End Security in einem Rückversicherungsunternehmen

Problemstellung

- Ist Zustand:
  - beschränkte Möglichkeiten des externen Zugriffs auf zentrale IT-Infrastrukturen
  - Eine umgesetzte Lösung: IPSec-VPN
  - Nicht alle Anwendungsfälle abgedeckt
- Firmenspezifische Sicherheitsanforderungen an Lösungen
- Restriktionen durch bestehende Infrastruktur
- Ziel:
  - Selektion alternativer technischer Lösungsalternativen

Einsatzszenarien

- 1: Ein Mitarbeiter der MR soll mit Hilfe des Firmenlaptops oder anderer VU(Versicherungsunternehmen)-Hardware von zu Hause aus oder unterwegs arbeiten können „wie im Büro“ (Teleworking). Wesentlicher Aspekt dabei ist, dass die Hardware Eigentum des VU ist und somit sämtlicher administrativer Aufwand sowie die Wartung auch durch das VU getragen werden. Es wird hingegen keine Aussage über die verwendeten Netze, Einschränkungen von Applikationen o.ä. getroffen.

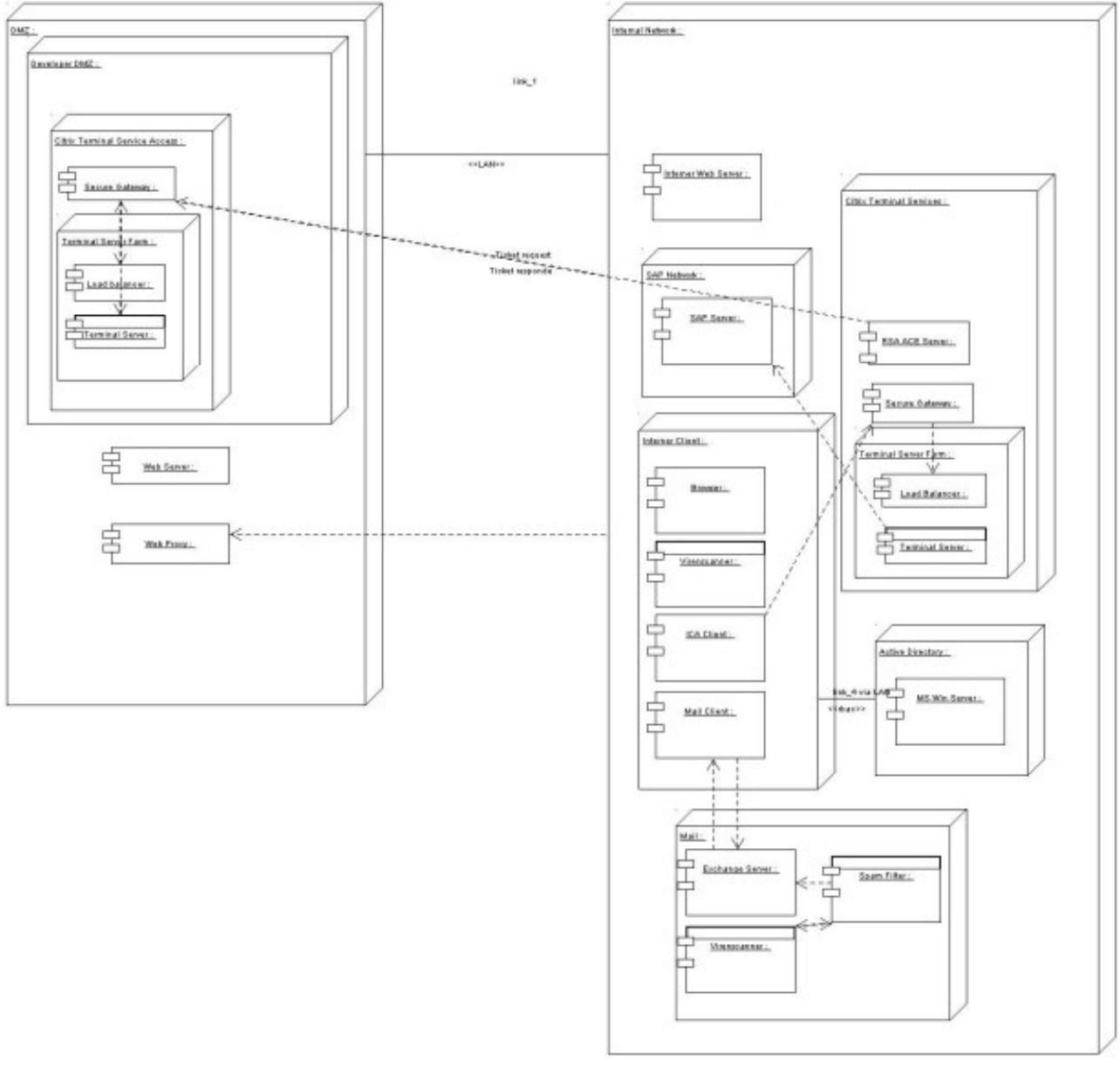
Einsatzszenarien

- 2: Ein Mitarbeiter soll von überall (jedem Gerät, jedem Standort) auf seine Mails, Termine und Kontakte zugreifen können (Mailaccess). In diesem Fall spielt die verwendete Hardware keine Rolle, es erfolgt aber eine Einschränkung betreffs der verwendeten Applikation.

Einsatzszenarien

- 3: Unternehmen bzw. externe Mitarbeiter, die nicht zur MR-Gruppe gehören, sind in Projekte oder andere Aufgabenstellungen bei des VU eingebunden. Sie sollen in dem Umfang Zugriff auf die von ihnen benötigten Ressourcen bei dem VU haben, wie dies für die Erfüllung ihrer Aufgabenstellung notwendig ist (Rolebased Access). In diesem Fall erfolgt weder eine Einschränkung auf Applikationen, Netze oder Ressourcen, sondern es wird implizit ein spezielles Datenberechtigungsmodell unterstellt.

IT Landschaft des VU



Relevante Rahmenbedingungen für einer RV

- Nationale Vorschriften
  - Branchenspezifische Vorschriften
  - Vorschriften in einem Versicherungskonzern
    - Z.B. MARisk (VA)
- Internationale Vorschriften
  - SOX
  - Basel II
  - Gramm Leach Bliley Act
- Notwendigkeit eines laufenden Risikomanagement

Untersuchte Technologien

- Umsetzung von End-to-End Security
  - Virtual Private Network (VPN)
    - IPSec-VPN
    - SSL-VPN
  - Terminal Server Technologie
  - Fernsteuerungslösungen
- Lösungen für spezifische Problemstellungen
  - Webbasierte Dienste
  - Weitere Lösungen auf Applikationsebene

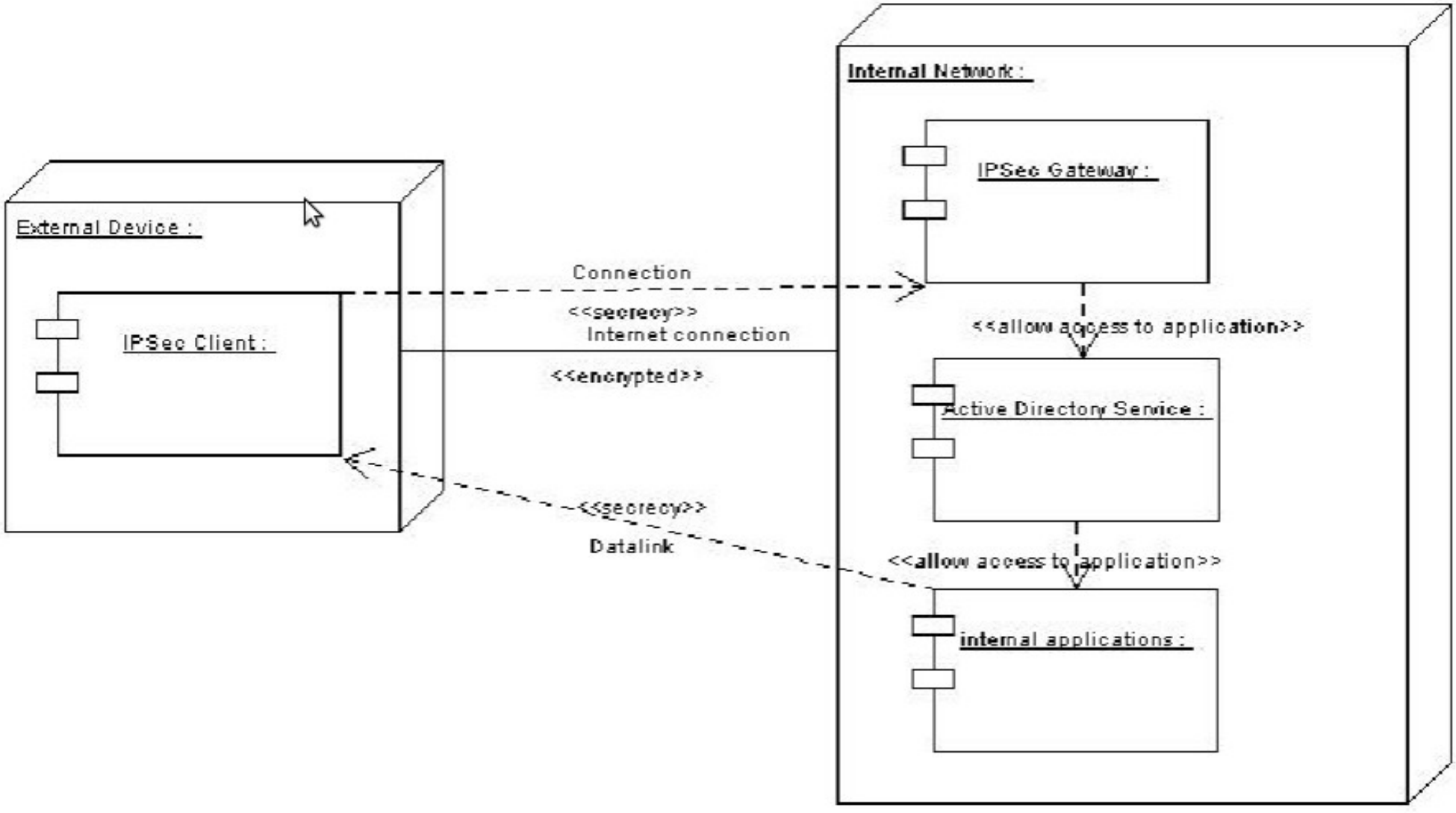
Modellierung auf Basis von UML

- Ziel: Bewertung der Technologien mit Hilfe von UMLsec
  - Probleme:
    - UMLsec gut bei der Entwicklung von sicherer Software
    - Untersuchung von Produkten mangelt es am Zugang zu den Sourcen
    - Komplexität der Produkte
  - Daher: Beschränkung auf die architekturelle bzw. physikalische Sichtweise
    - Darstellung mit Hilfe von Deployment-Diagrammen
    - Verifizierung im Viki-Framework
    - Beispiele: VPN, Terminal Server
-

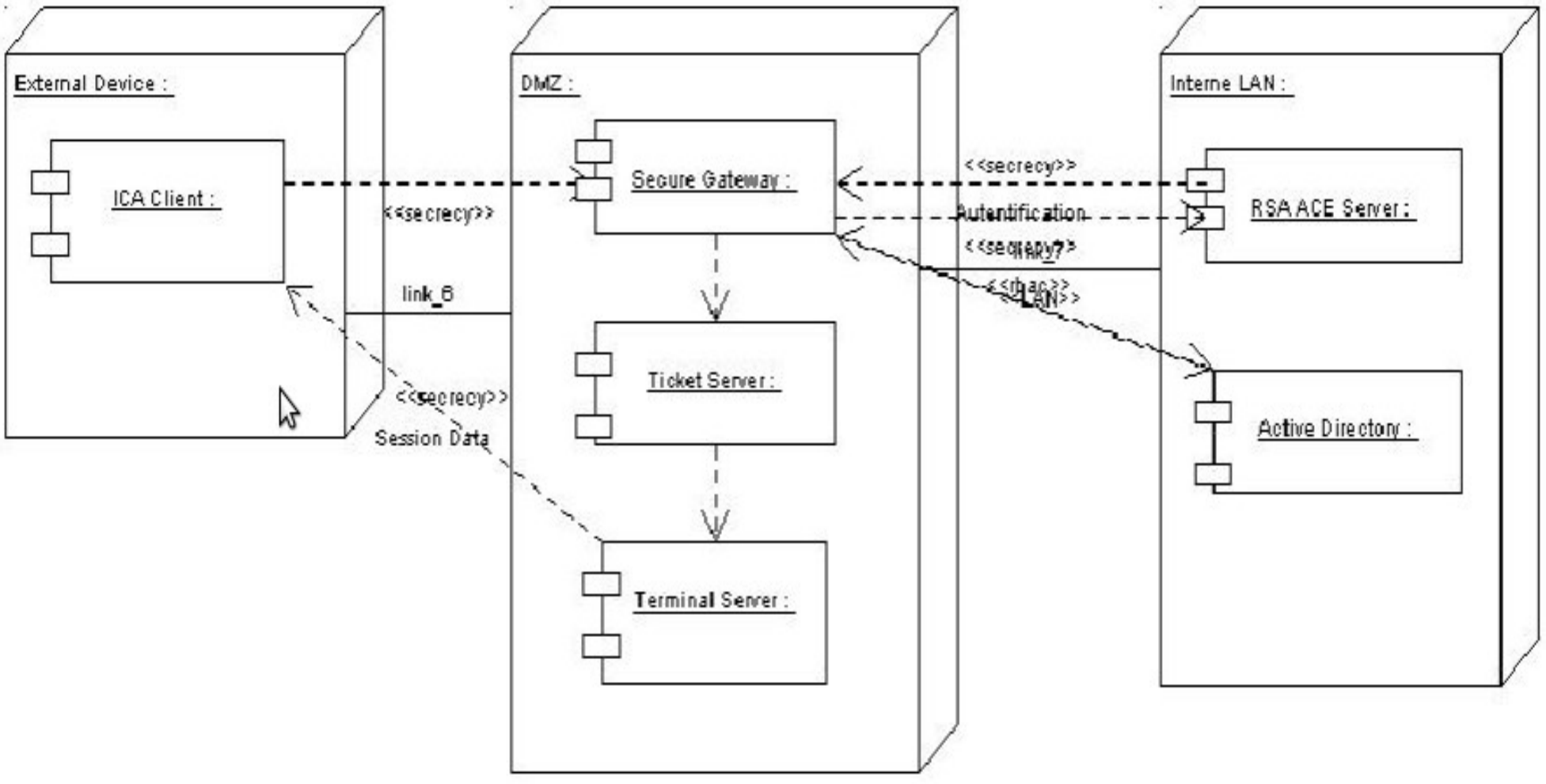
Beispiele: Angreifermodell

- Erweiterung der UML um Stereotypen, Tags und Constraints
- Definition eines Angreiferprofils: Man-in-the-middle bei externen Verbindungen
- Szenarien bzw. Verbindungsmöglichkeiten des Angreifers haben Einfluss auf seine Fähigkeiten:
  - Internet by Mobile Internet {read}
  - by WLAN {read}
  - Telecommunication Internet by Cable {delete, read, insert}

SSL / VPN Modell



Terminal Server Modell



Ergebnis

- Mit Toolunterstützung können keine Sicherheitslücken festgestellt werden
- Und: Keine Aussage über ein einzelnes Produkt getroffen
- Folge: anderes Konzept zur Risiko- und Nutzenbewertung notwendig
  - Threat Modelling (Teil der UML 2.0 Spezifikation
  - Kosten Nutzen Analysen
  - Balanced Score Cards

Fazit

- Methoden der UMLsec erlauben Sicherheitsanalyse, können aber nur bei umfassenden Informationen der zu Grunde liegenden Software sinnvoll genutzt werden.
- Verwendung von UML kann auf Management-Ebene verwendet werden. Nutzung im Risikomanagement (Threat-Modelling).
- Keine allgemeine Aussage über die Vorteilhaftigkeit von Technologien bzw. einzelnen Produkten, abhängig vom Anwendungsfall