LEHRSTUHL 14
SOFTWARE ENGINEERING

Willkommen zur Vorlesung

Methodische Grundlagen des Software-Engineering

im Sommersemester 2011

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

27. Software Architektur Anwendungsbeispiele

Modellierung und Verifikation von mehrschichtigen Sicherheitsarchitekturen: Eine Bankapplikation

Mehrschichtige Sicherheitsprotokolle

- 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

client authenticity

confidentiality, integrity, server authenticity

= ?

confidentiality, … + client authenticity

Bank-Anwendung

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

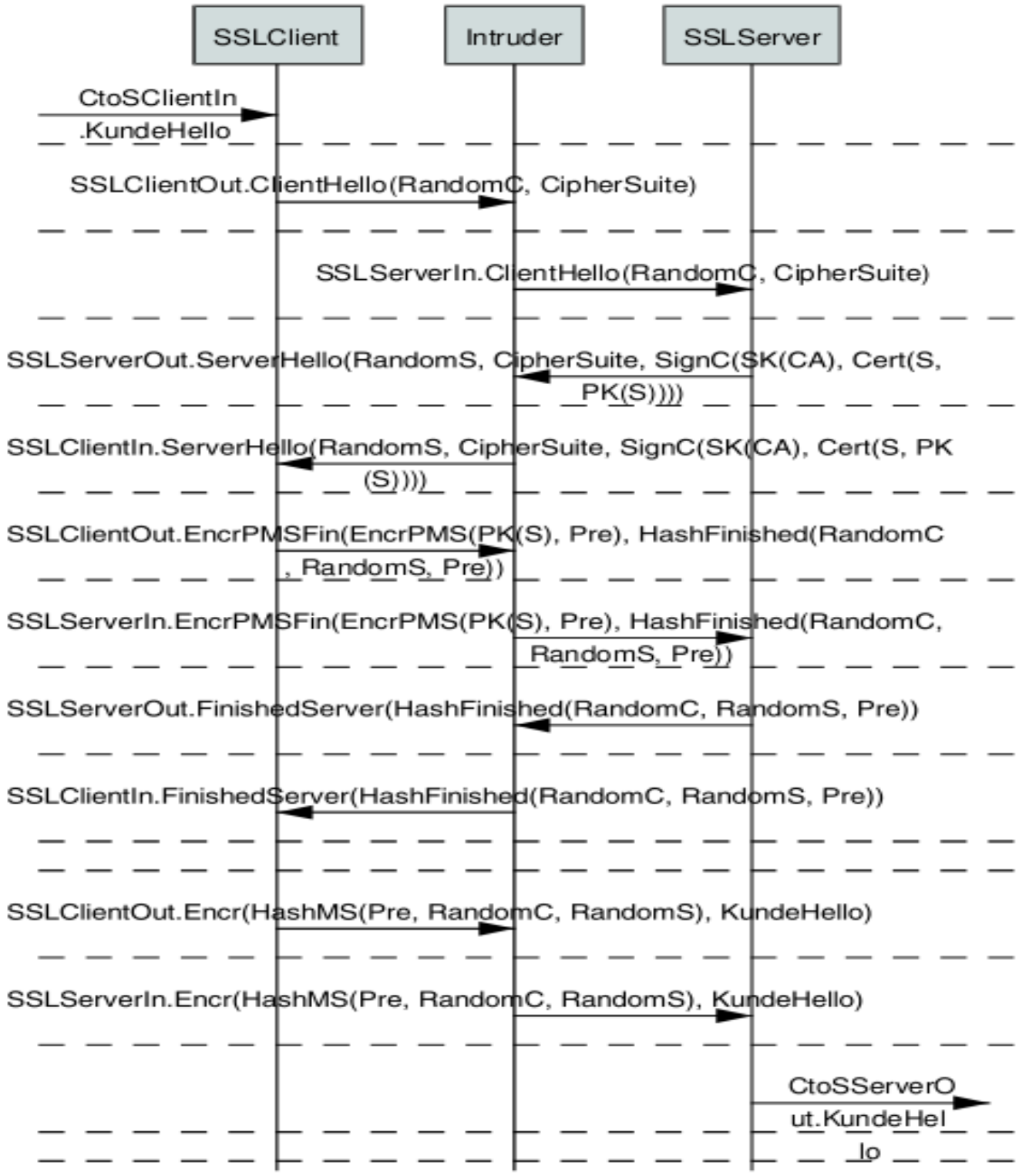
Bank-Anwendung

- 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

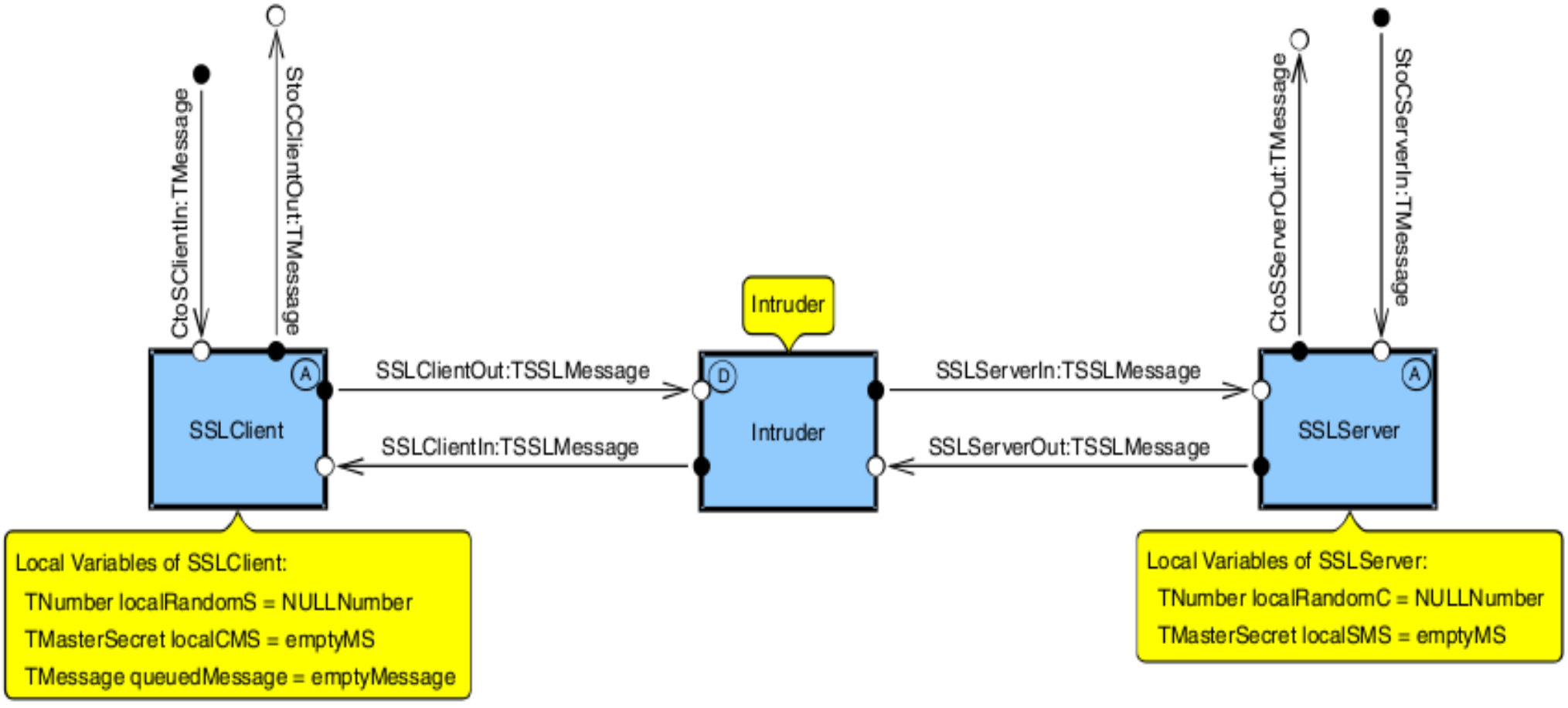
Angreifermodell

- Anpassung des Angreifermodells um SSL Sicherheitseigenschaften zu berücksichtigen.
- Begründung, dass das angepasste Angreifermodell bezogen auf das top-level-Protokoll genau so stark ist, wie das generische Angreifermodell bezüglich des Protokoll-Stacks.
- Verifizkation des top-level-Protokoll in Bezug auf das angepasste Angreifermodell.
- Impliziert Verifikation des Protokoll-Stack.

SSL Protokoll mit Angreifer



SSL Protokoll mit Angreifer: Kommunikationsarchitektur



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 der Sicherheitsanalyse
  - 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 der Sicherheitsanalyse
  - 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. Der Webserver 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.

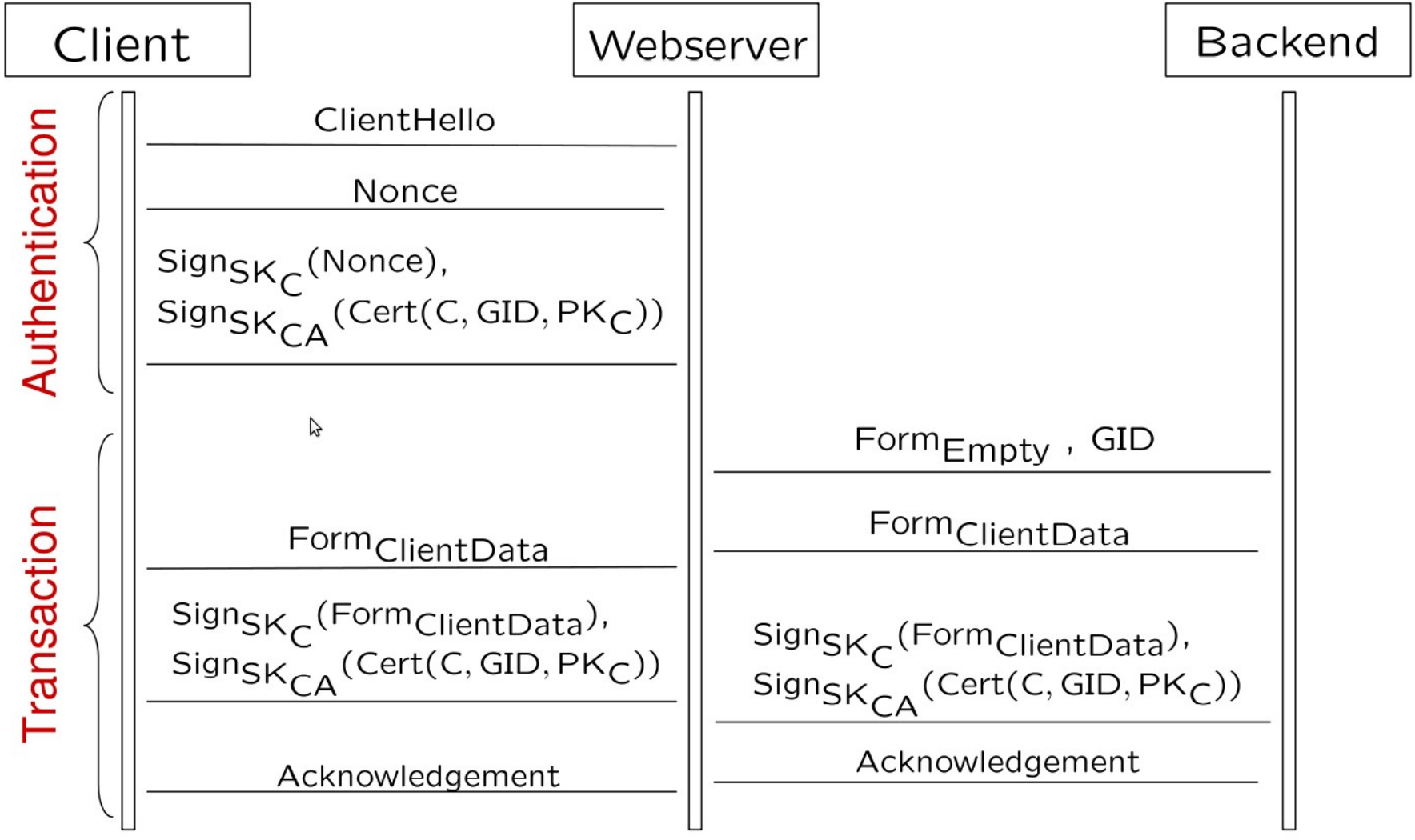
Client Authentifizierungsprotokoll

- 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.

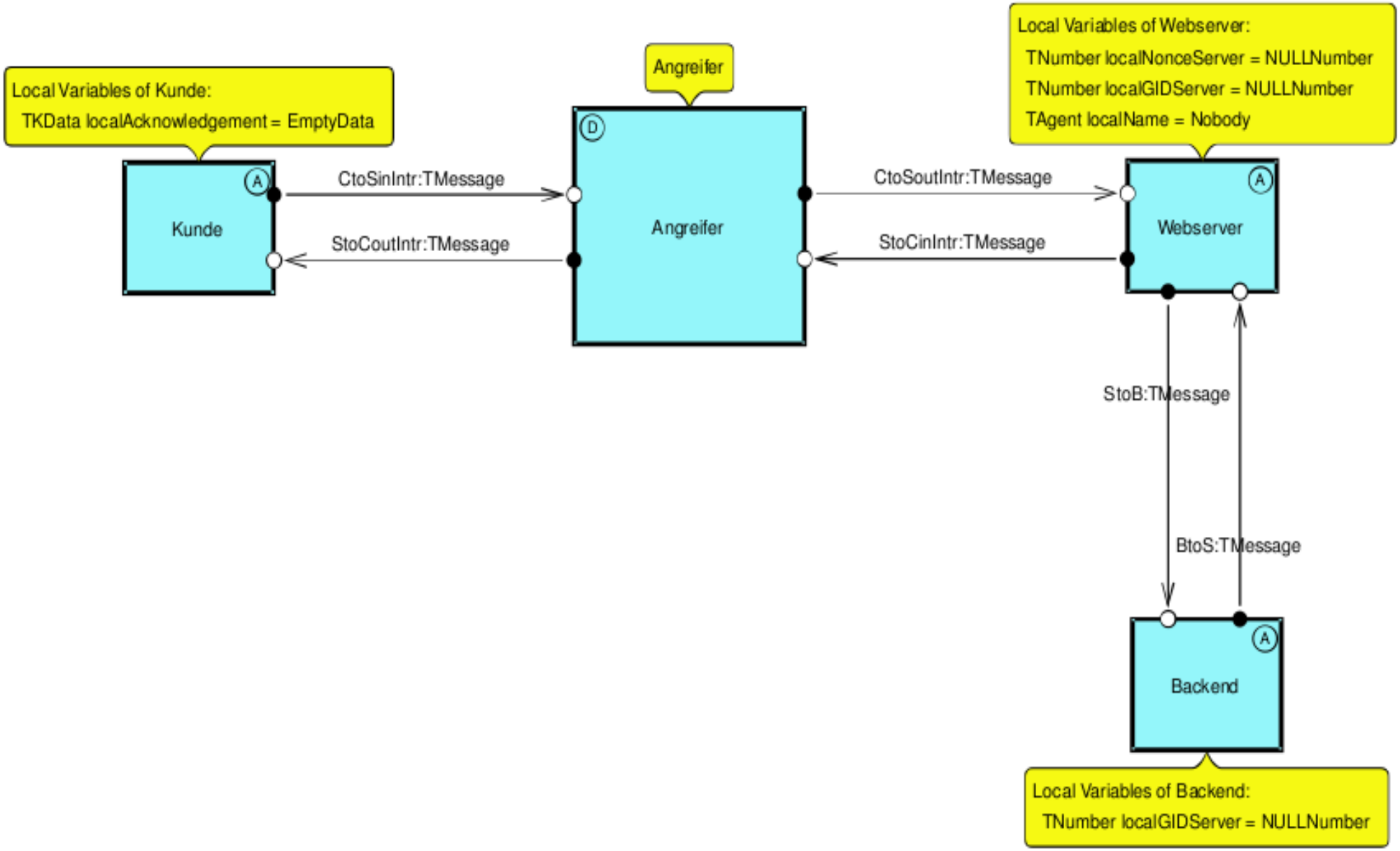
Client Authentifizierungsprotokoll

- 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: Kommunkationsarchitektur



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.