

Methodische Grundlagen des Software Engineering - Übung 12

12 UML und UMLsec

Abgabe der Hausaufgaben am Anfang der jeweiligen Präsenzübung am 05.07.2011 bzw.
06.07.2011.

12.1 Anwendungsfalldiagramm

12.1.1 Modellieren sie folgenden Prozess als UML Anwendungsfalldiagramm

Es geht um den generellen Ablauf in einem italienischen Fastfood Restaurant. Der Kunde betritt das Restaurant und überlegt sich anhand der aushängenden Karten was er essen will. Dann geht er zur Kasse und nennt seinen Wunsch. Die Kasse berechnet daraufhin den Preis und nennt ihn. Der Kunde bezahlt und bekommt einen Beleg über seine Bestellung. Mit diesem Beleg geht der Kunde zur Theke und übergibt den Beleg dort. Der dortige Angestellte prüft ob die Bestellung verfügbar ist. Wenn sie es nicht ist, bekommt der Kunde seinen Beleg zurück, geht damit zur Kasse und bekommt sein Geld zurück. Danach kann er gehen oder neu bestellen. Ist die Bestellung durchführbar wird sie an die Küche gegeben. Dort wird das Essen gekocht. Ist es fertig wird es bereitgestellt und ein Signal für die Theke gegeben. Diese nimmt das Essen an und übergibt es an den Kunden. Der sucht sich dann einen Platz, isst sein Essen und verlässt das Restaurant.

12.2 Aktivitätsdiagramm

12.2.1 Modellieren sie Prozess aus 12.1.1 als UML Aktivitätsdiagramm

12.3 Verteilungsdiagramm

12.3.1 Modelliere folgendes als Verteilungsdiagramm. Substituiere Component(UML 1.5) durch Artifact(UML 2.3).

Für das Szenario aus 12.1.1 plant der Betreiber nun eine modernisierte IT Unterstützung. Dabei sollen Tabletpcs für die Kasse, die Theke und die Küche sowie ein USB Stick basiertes Token für den Beleg zum Einsatz kommen. Die KasseApp nimmt dabei die Bestelldaten auf und überträgt diese per WLAN an die ThekenApp. Auf das Token wird nur ein zufällig erzeugter Identifier gespeichert. Das Token wird dazu per Kabelverbindung mit dem Kassentablet verbunden. Auf die selbe weise wird der Token an das Thekentablet angeschlossen und dort ausgelesen. Die ThekenApp ruft dann anhand des Identifiers die zugehörigen Daten auf. Wird die Bestellung angenommen, wird der Identifier auf dem Token gelöscht und es werden die Daten an das Kuechentablet weitergeleitet. Dort werden sie von der KuechenApp angezeigt. Das Kuechentablet ist per Lan mit dem Kassentablet verbunden. Ist das Essen fertig sendet die KuechenApp der ThekenApp ein Signal. Solange der Identifier auf dem Token noch gespeichert ist kann man sich an der Kasse auszahlen lassen.

12.4 Fair exchange

- 12.4.1 *Erweitere das Aktivitätsdiagramm aus 12.2.1 um die fair exchange stereotypen und tagged values*
- 12.4.2 *Erweitere das Aktivitätsdiagramm aus 12.1.1 um die fair exchange stereotypen und tagged values*
- 12.4.3 *Ist fair exchange erfüllt?*
- 12.4.4 *Es kommt zur bisherigen Prozessbeschreibung folgende Aussage hinzu: "Wenn ein Beleg zu spät zurück gegeben wird an der Kasse wird kein Geld ausgezahlt". Nehme diese Aussage in das Aktivitätsdiagramm aus 12.4.2 auf.*
- 12.4.5 *Ist fair exchange noch erfüllt?*

12.5 Secure Links

- 12.5.1 *Erweitere das Modell aus 12.3.1 um die UMLsec Secure Links Stereotypen. (Kabel als <<Wire>>, Wlan als <<Wlan>>). Substituiere Component(UML 1.5) durch Artifact(UML 2.3)*
- 12.5.2 *Ist Secure Links erfüllt? Nimm dabei den Standardangreifer an und setze <<Wire>> mit <<Lan>> und <<Wlan>> mit <<Internet>> gleich. Wenn es nicht sicher ist, verändere das Modell so, das Secure Links eingehalten wird.*

Hausaufgabe

12.6 Fair exchange

12.6.1 Modelliere folgendes als Aktivitätsdiagramm. Sorge dafür das ein "fair exchange" stattfindet und das Diagramm entsprechend annotiert ist.

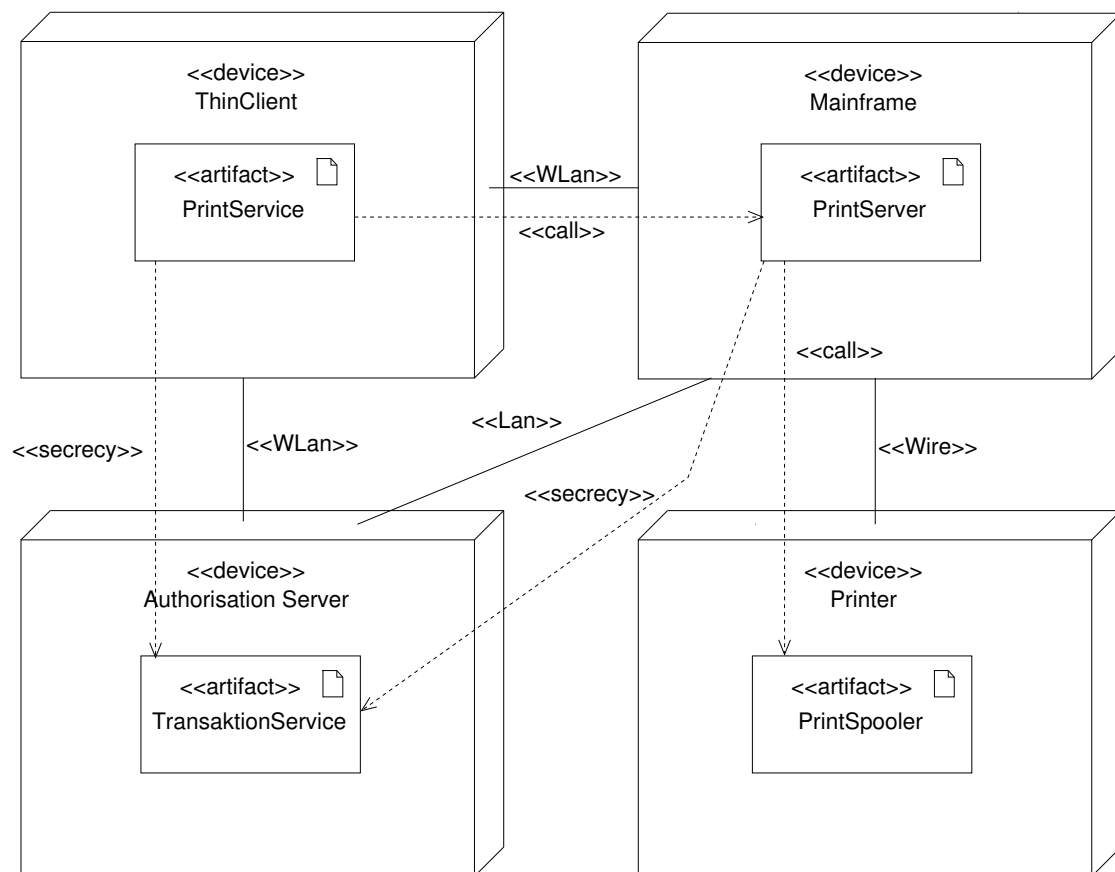
Ein Kinokunde möchte sich einen Film anschauen und dafür reservieren. Dafür verbindet er sich zu dem Reservierungsserver des Kinos und fordert eine Liste der Vorstellungen an. Der Reservierungsserver liefert diese. Nachdem der Kunde eine Vorstellung gewählt hat, zeigt der Server zudem die Sitzplatzbelegung dieser Vorstellung an. Der Kunde wählt nun seine Wunschplätze und startet damit die Reservierung. Auf Seiten des Reservierungsservers wird nun geprüft ob die Wunschplätze noch frei sind. Wenn nicht wird eine Fehlermeldung generiert und der Kinokunde bricht seine Reservierung ab. Sind die Plätze frei, generiert der Reservierungsserver ein Bestellformular welches vom Kunden ausgefüllt wird. Nach dem Übersenden der Daten fragt der Reservierungsserver nach einer Bestätigung. Hat sich der Kinokunde es nun anders überlegt, storniert er die Reservierung und der Reservierungsserver löscht die Reservierung. Bestätigt der Kunde die Reservierung wird die Reservierung fixiert und das Geld abgebucht. Auf Seiten des Servers wird nun bis 30 Minuten vor Beginn der Vorstellung gewartet. Dann wird geprüft ob die Karten abgeholt wurden. Wurden sie nicht abgeholt, wird das Geld zurück gebucht und die Reservierung gelöscht.

Der Kunde merkt sich die Reservierungsnummer und holt irgendwann die Karten ab. Die Kasse ruft nun anhand der Reservierungsnummer die Reservierung vom Reservierungsserver ab und fordert eine Authentifizierung vom Kunden. Dieser zeigt seinen Ausweis und die Kasse prüft ihn. Wenn der Ausweis nicht zur Reservierung passt wird der Vorgang abgebrochen, ansonsten werden die Karten übergeben und dem Reservierungsserver mitgeteilt, das die Reservierung eingelöst wurde. Dieser speichert das.

Der Kunde nimmt die Karten an und schaut dann den Film. 4 P

12.7 Secure Links

12.7.1 Ist folgendes Verteilungsdiagramm gemäß Secure Links und dem Default Attacker sicher? Wenn nicht, was muss angepasst werden? Setze `<<Wire>>` mit `<<Lan>>` und `<<Wlan>>` mit `<<Internet>>` gleich.



1 P