Methodische Entwicklung sicherer CORBA-Anwendungen

Jan Jürjens

Software & Systems Engineering
Informatik, TU München

juerjens@in.tum.de

http://www.jurjens.de/jan

Sicherheit

Unternehmen sind zunehmend abhängig von Computernetzwerken.

Angriffe bringen ein zunehmendes Risiko.

Vernetzte Systeme können anonym und aus sicherer Distanz angegriffen werden.

Vernetzte Computer müssen sicher sein.

Probleme

Viele Schwachstellen in Entwürfen sicherheitskritischer Systeme, manchmal erst nach Jahren gefunden.

Beispiel (1997):

NSA Hackerteam bricht in U.S. Department of Defense Computer und U.S. Stromversorgungssystem ein.

Simuliert Stromausfälle und Notrufausfälle.

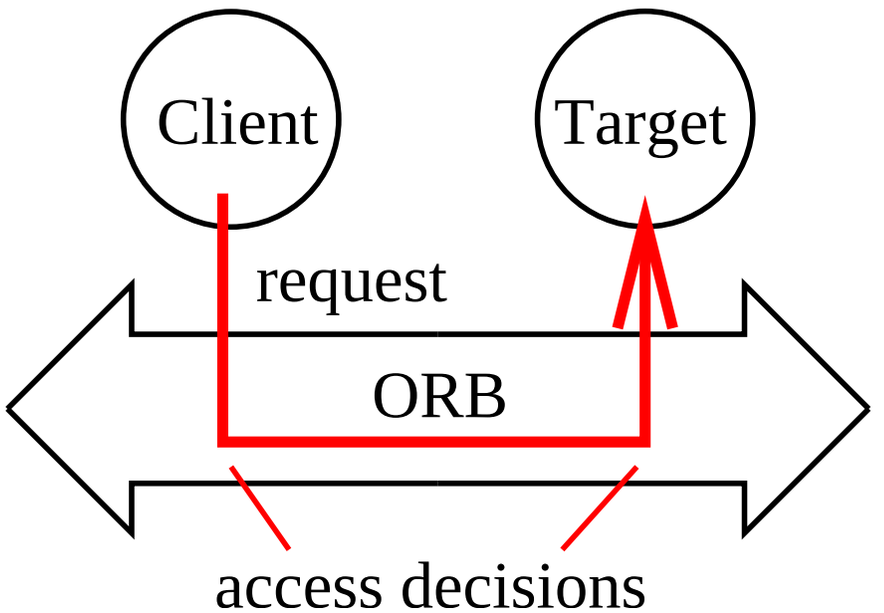
Gründe

- Korrekter Entwurf sicherer Systeme schwierig.
- Entwicklern fehlt oft Sicherheitsausbildung.
- Sicherheitserwägungen im Nachhinein.
- Kann Sicherheitsmechanismen nicht "blind" einsetzen (meiste Angriffe nutzen Systemkontext).

CORBA Zugangskontrolle

Zugriffssteuerungspolitik für Objektaufrufe kontrolliert Zugang eines Client zu bestimmtem Objekt über bestimmte Methode.

Realisiert durch Object Request Broker (ORB) und Security Service.



Zugangsentscheidungsfunktionen

Zugangsentscheidungsfunktionen entscheiden über Zugangserlaubnis. Abhängig von

- aufgerufener Operation,
- Privilegien des Client-Owners,
- Kontrollattributen des Zielobjektes.

Zugangskontrolle mithilfe von Schutzgebieten (statt einzelner Objekte).

Ausführungszusammenhang

Zugang hängt vom Ausführungskontext mit Credentials ab:

- eigene Credentials,
- von aufrufenden Objekten empfangene Credentials,
- an aufzurufene Objekten zu schickende aufrufende Credentials.

Aktueller Ausführungskontext eines Programmes in **Current** Objekt, von ORB gelesen.

Zugangskontrollpolitiken

Methode kann Rechte bei aufrufendem Objekt fordern:

g "get" Recht kontrolliert Zugang zu Methoden, die Informationen zurückgeben.

s "set" Recht kontrolliert Zugang zu Methoden, die Informationen im aufgerufenen Objekt ändern.

u "use" Recht kontrolliert Zugang zu Methoden, die Aufgabe ausführen lassen.

m "manage" Recht kontrolliert Zugang zu normalen Benutzern unzugänglichen Methoden.

Zugangsentscheidungsobjekte

Access decision objects (ADO's) setzen Politiken durch.
Können auf Anwendungsebene definiert werden.

ADO's erlauben differenzierte Sicherheitspolitiken. Aber:

- Zugangserlaubnis abhängig von Ausführungs-zusammenhang und Laufzeitparametern.
- Authentifizierung diffizil.
- Indirekte Zugangserlaubnis mittels Credentials.

$\rightsquigarrow$ Schwierig zu sehen, welche Objekte Erlaubnis erhalten.

Lösung: UML

Unified Modeling Language (UML):

- visuelles Modellieren OO Systeme
- verschiedene Ansichten eines Systems
- hoher Abstraktionsgrad möglich
- de-facto Industriestandard (OMG)
- Standard-Erweiterungsmechanismen

Formale Semantik für automatisierte Analyse, Konsistenztests, beweiskräftige Sicherheitsanalysen. . .

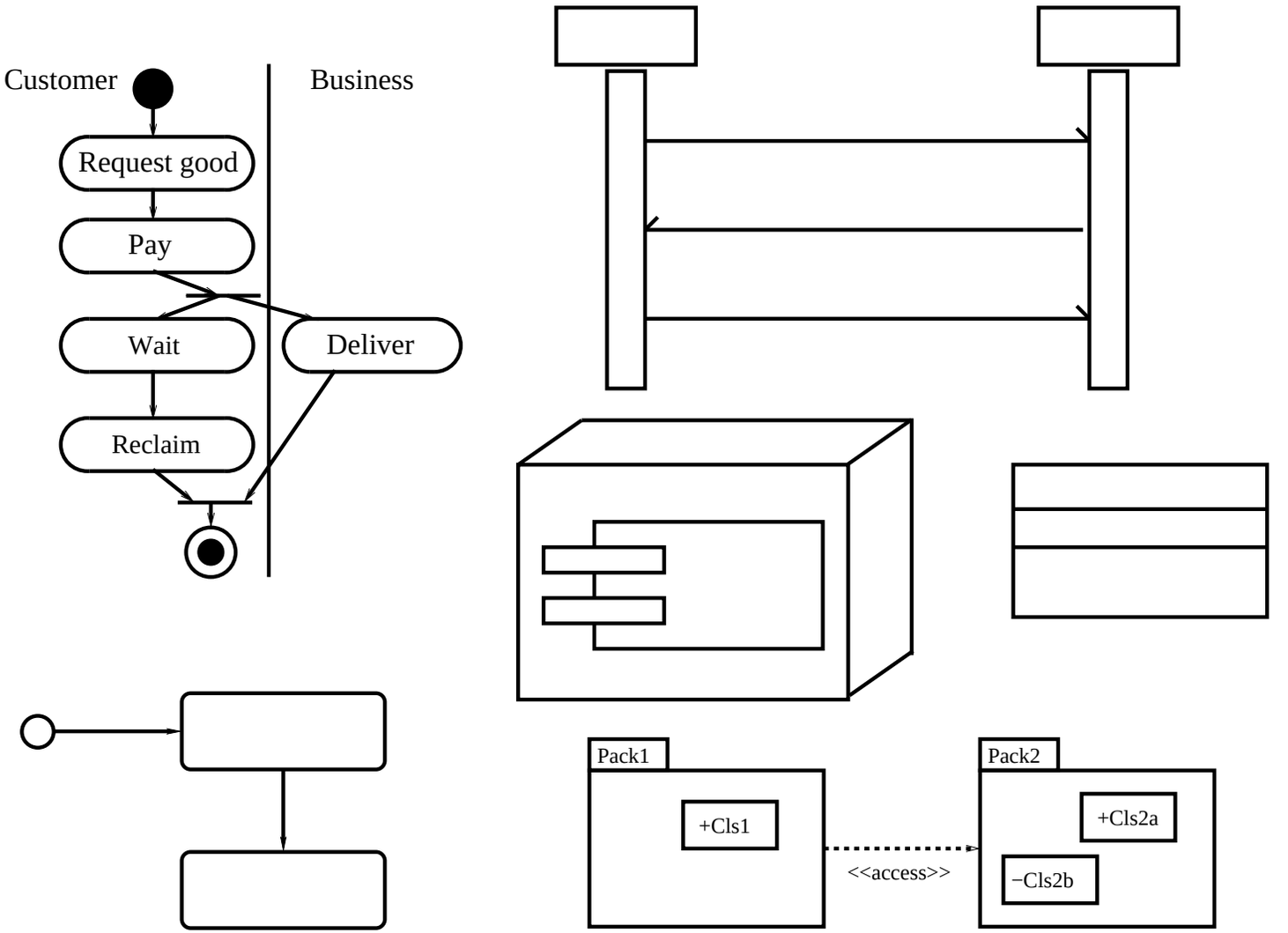
UMLsec

Bislang einzige Erweiterung für Entwicklung sicherer Systeme.

- UML Spezifikationen auf Schwachstellen im Entwurf auswerten
- etablierte Regeln des Security Engineering ausdrücken und
- Entwicklern ohne Sicherheitsausbildung zur Verfügung stellen
- Sicherheit von frühen Entwicklungsphasen an, im Systemkontext
- kosten-effiziente Zertifizierungen

Hier: korrekte Verwendung der CORBA Zugangskontrolle

UML: Überblick



Benutzter Teil der UML

Aktivitätsdiagramm: Kontrollfluß zwischen Systemkomponenten

Klassendiagramm: Klassenstruktur des Systems

Sequenzdiagramm: Interaktion zwischen Komponenten durch Nachrichtenaustausch

Zustandsdiagramm: dynamisches Komponentenverhalten

Einsatzdiagramm: Komponenten in physikalischer Umgebung

Package: Systemteile zusammenfassen

Das UMLsec Profil

Häufige Sicherheitsanforderungen als Stereotypen mit Tags (Vertrauchlichkeit, Integrität, . . .).

Assoziierte Bedingungen evaluieren Spezifikationen und zeigen mögliche Schwachstellen.

Überprüft, dass formulierte Sicherheitseigenschaften Sicherheitspolitik durchsetzen.

Überprüft, dass UML Spezifikation Sicherheits-eigenschaften erfüllt.

Z.B.: Sicherheitsattribute von Kommunikationsbeziehungen

Validierung: Präzise Semantik für Verhalten.

Entwicklungsprozess

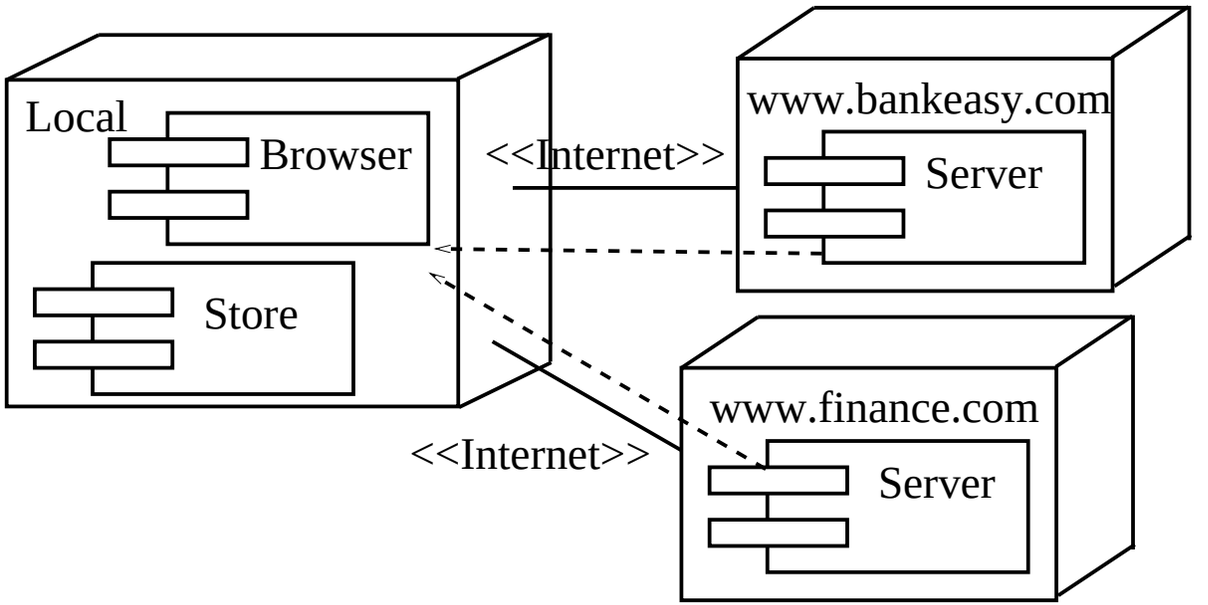
(1) Zugangskontrollanforderungen für kritische Objekte formulieren.

(2) ADO's mit passenden Zugangskontrollbedingungen definieren.

(3) Überprüfen, dass ADO's Objekte hinreichend schützen.

(4) Überprüfen, dass Zugangskontrolle konsistent mit Funktionalität.

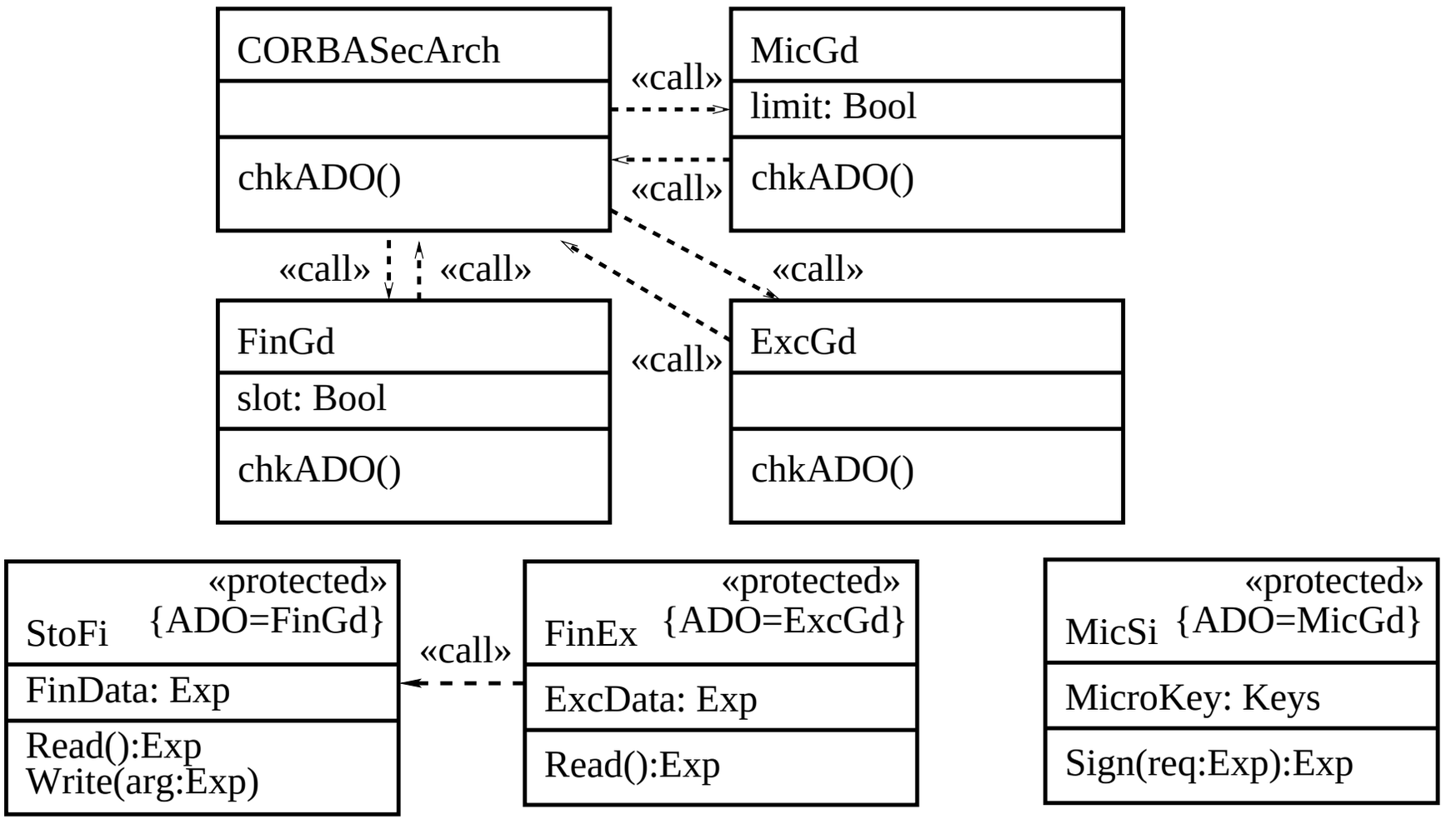
Beispiel: Finanzanwendung



Dienste von Internetbank und Finanzberater für lokalen Benutzer. Objekte brauchen Privilegien (Schritt 1).

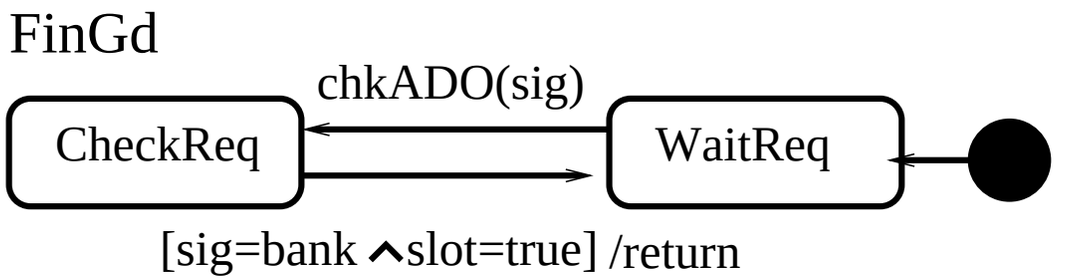
- Objekte von Bank lesen und schreiben Finanzdaten zwischen 13.00 und 14.00 Uhr.
- Objekte von Finance benutzen Micropayment Schlüssel fünf Mal pro Woche.

Finanzanwendung: Klassendiagramm

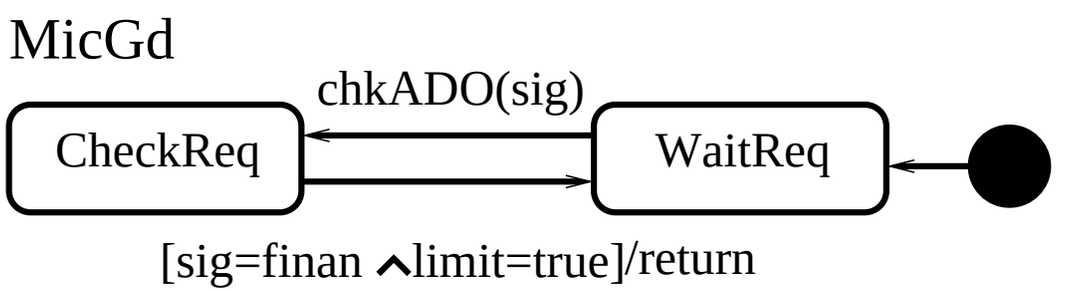


Finanzanwendung: ADO's (step 2)

slot wahr zwischen 13.00 und 14.00 Uhr.



limit wahr bis Zugang fünf Mal gewährt.



Finanzanwendung: Validerung

Verifizieren:

ADO's geben hinreichenden Schutz (Schritt 3).

Zugangskontrolle konsistent mit Funktionalität (Schritt 4).

Werkzeugunterstützung

Zeichenwerkzeug (Rational Rose, . . .)

mit XMI (XML Metadata Interchange) zu:

Analysewerkzeug (AutoFocus)

- Testsequenzgenerierung
- Verifikation
- Codegenerierung

Weiterhin in Arbeit: Standardisierung

Security im Software Engineering

Anwendungsprojekte mit UMLsec und AutoFocus:

- Projekt FairPay des Bundesministerium für Wirtschaft
- Common Electronic Purse Specifications
- Java Security
- elektronisches Ausschreibungssystem
- elektronische Geldbörse für Palmtop
- Internet-Sicherheitsprotokolle
- verschiedene Projekte mit Industriepartnern

http://wwwbroy.in.tum.de/~secse