Java-Security

Teil I

Vorlesung ***Firewalls und andere IT-Sicherheitsmechanismen***

Kai Schmitz-Hofbauer

7. Februar 2003

Inhalt Teil I

- Einleitung
- Sicherheitsphilosophie
- Sprachsicherheit
- Sicherheitskomponenten der VM
- Das Sandbox-Modell und dessen Weiterentwicklung
- Security- API und Kryptographie

Einleitung

- Java ist eine objektorientierte, plattformunabhängige Programmiersprache.
- Zwei Arten von Java-Programmen : Applikationen und Applets
  - ⋆ Applikationen sind eigenständige Programme.
  - ⋆ Applets werden innerhalb einer HTML-Seite dargestellt und unter der Kontrolle eines Web-Browsers ausgeführt.
  - ⋆ Applets werden in der Regel über das Internet geladen
    $\Rightarrow$ besondere Sicherheitsvorkehrungen notwendig.
- Java-Quellcode wird beim Compilieren in Bytecode übersetzt.
- Bytecode wird von einer virtuellen Maschine interpretiert und ausgeführt.

Sicherheitsphilosophie

- Sun verfolgt bei Java die Philosophie *Sicherheit durch Offenheit*.
- Sicherheitsmodell, Konzepte, Spezifikationen, Implementierungsdetails und sogar der Quellcode von Java wurden veröffentlicht.
- Durch die so geschaffene Transparenz können Sicherheitslücken schnell lokalisiert und beseitigt werden.
- Von *Security by Obscurity* (Sicherheit durch Verschleierung) kann keine Rede sein.

Sprachsicherheit

- Verzicht auf Pointer
- strenge Typisierung
- Initialisierung der Variablen vor ihrer Verwendung mit einem gültigen Default-Wert
- voll-automatisches Speicher-Managment
  - ⋆ Überwachung der Grenzen der Speicherbereiche während der Laufzeit
  - ⋆ insbesondere strenge Überprüfung von Array- und Stringgrenzen
    $\Rightarrow$ $BufferOverflow$ nicht möglich
  - ⋆ Garbage Collection kümmert sich um die Speicherfreigabe
- $Final$ deklarierte Klassen und Methoden können keine Unterklassen bilden oder überschrieben werden.

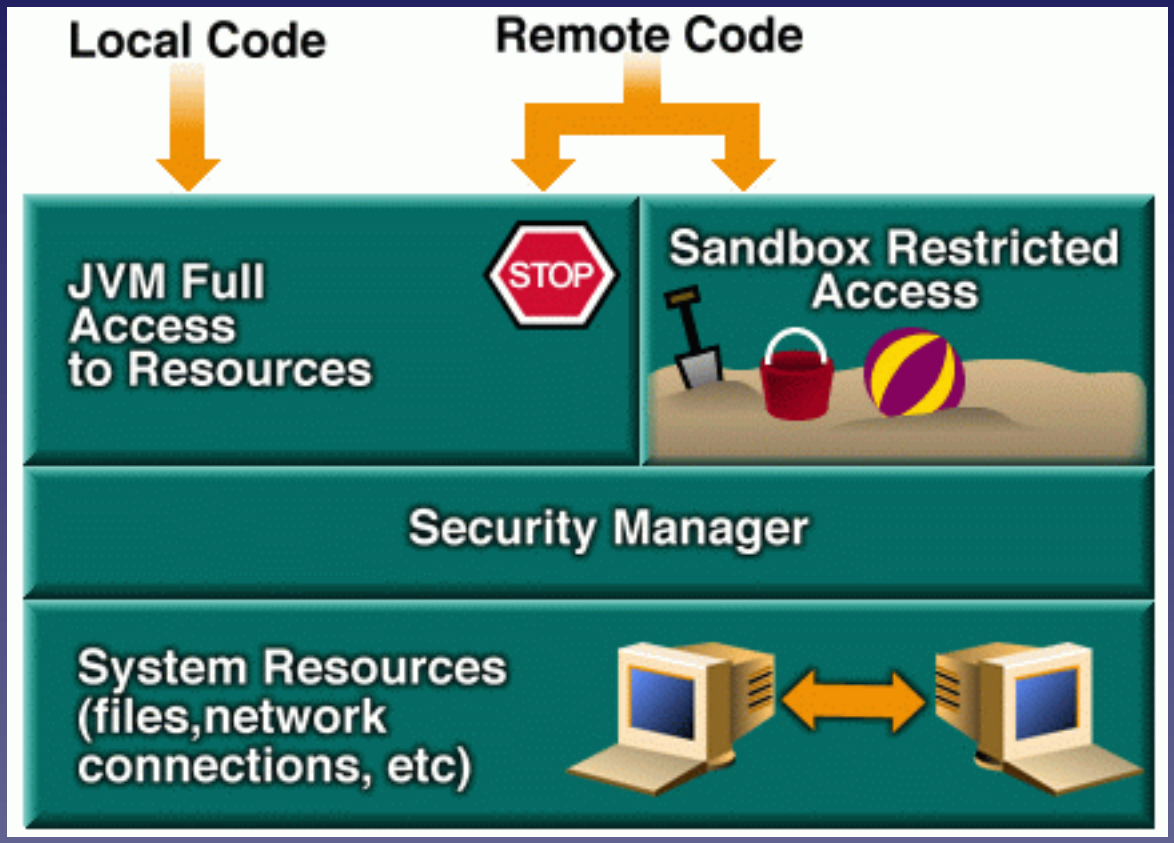
Sicherheitskomponenten der VM (1/2)

- Der Bytecode-Verifier
  - ⋆ stellt vor Ausführung sicher, dass der Bytecode den Spezifikationen entspricht.
- Der Classloader
  - ⋆ für das Lokalisieren und Laden von Klassen zuständig
  - ⋆ stellt sicher, dass Klassen aus den Paketen **java.*** nicht über das Netzwerk geladen werden
  - ⋆ richtet Namensräume für Klassen gleicher Herkunft ein
  - ⋆ Klassen unterschiedlicher Herkunft können nicht auf den gleichen Namensraum zugreifen.

Sicherheitskomponenten der VM (2/2)

- In Java darf nicht jedes Programm auf jedes Betriebsmittel zugreifen.
  - ⋆ Applets (aus dem Internet) haben nur eingeschränkte Zugriffsrechte.
- Der SecurityManager
  - ⋆ überwacht den Zugriff auf kritische Ressourcen wie z. B. Dateien, Verzeichnisse oder Netzwerkports.
  - ⋆ Darf auf ein gewünschte Betriebsmittel nicht zugegriffen werden, so löst der SecurityManager eine SecurityException aus.
  - ⋆ Ansonsten gestattet er Zugriff auf das Betriebsmittel.

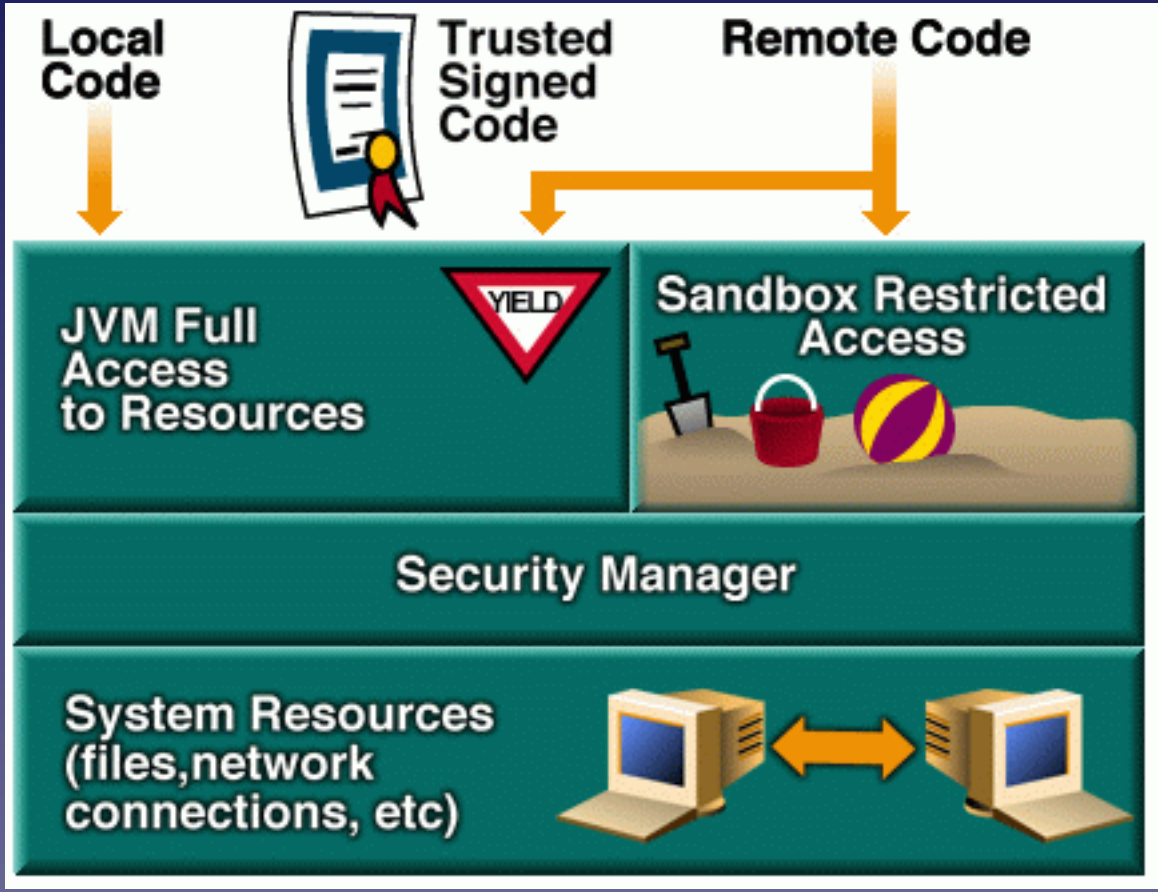
Das Sandbox-Modell aus Java 1.0



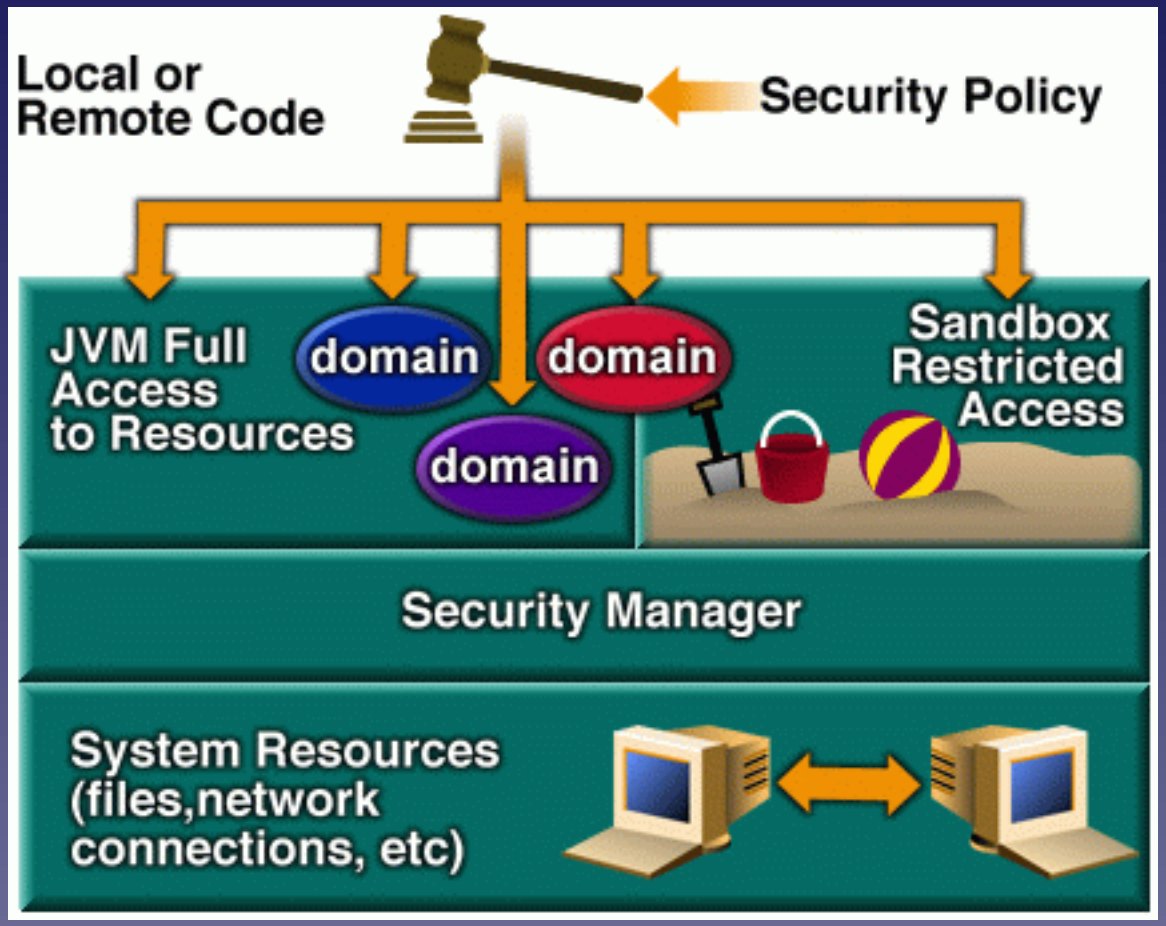
Verbote für entfernten Code

- Zugriff auf lokale Dateien und Verzeichnisse des Gastrechners
- Aufnehmen von Netzwerkverbindungen zu anderen Computern außer dem Heimatserver
- Ausführen von Programmen, die sich auf dem lokalen Rechner befinden
- Laden von nativen Bibliotheken
- Beenden der virtuellen Maschine
- Zugriff auf Informationen über das Java-Home-Verzeichnis, den Java-Klassenpfad, den Benutzernamen, das Home- und das Arbeitsverzeichnis des Anwenders.

Erweiterung des Sandbox-Modelles in Java 1.1



Weiterentwicklung des Sandbox-Modelles in Java 2



Security- API und Kryptographie

- Evtl. Notwendigkeit von Sicherheitsmechanismen auf höheren Ebenen bei Entwicklung eigener Java-Programme ⇒ Security-API (`java.security.*`)
- Bestandteil der Security-API ist die *Java Cryptography Architecture* (JCA) ⇒ Grundgerüst für die Benutzung und Entwicklung krypthographischer Verfahren
- *Java Cryptography Extension* (JCE) enthält Implementierungen der wichtigsten kryptographischen Verfahren (`javax.cryto.*`, `javax.security.*`)
-

`MessageDigest`	Berechnung von Hash-Werten
`Signature`	Erzeugung und Verifikation digitaler Signaturen
`KeyPairGenerator`	Generierung von (public- und private-) Schlüsselpaaren
`Cipher`	Ver- und Entschlüsselung

Vielen Dank für Ihre Aufmerksamkeit !

Fragen ?

Kai.Schmitz-Hofbauer@gmx.de