

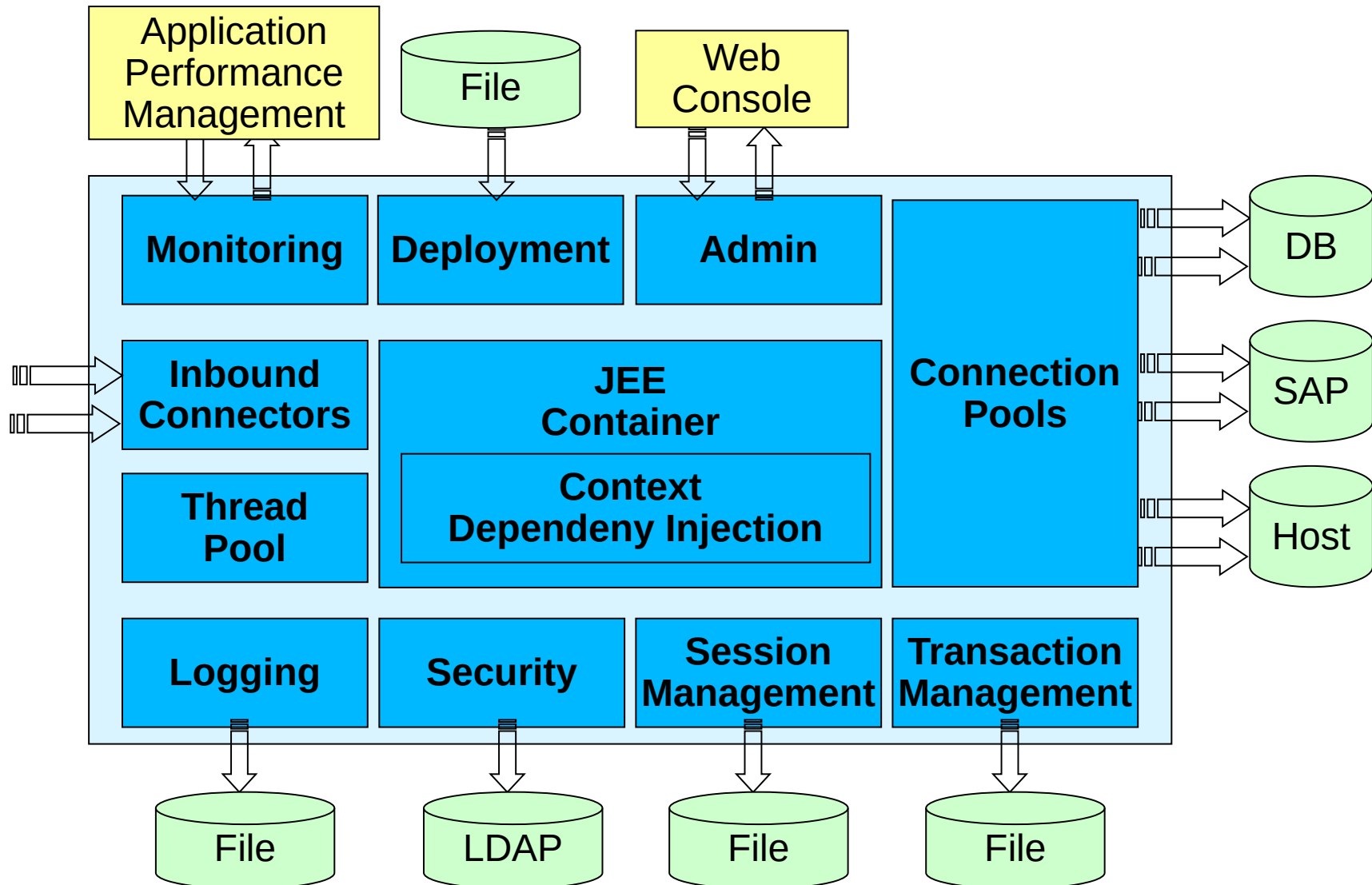
# Einführung in die Java / Jakarta Enterprise Edition

## Kapitel 2 – Der Applikationsserver

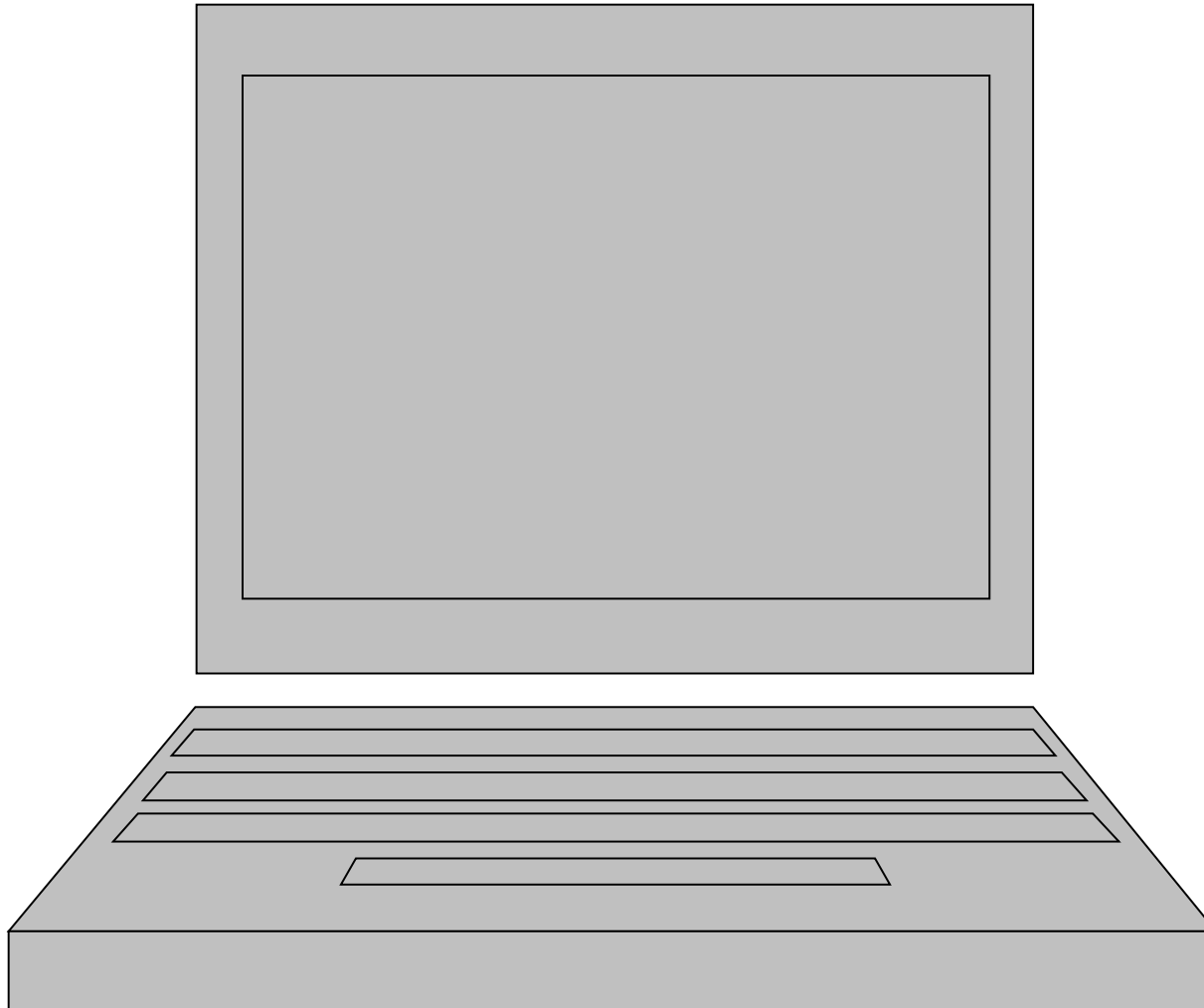
2.1

## **ARBEITSWEISE**

- Die Funktionen, die ein Applikationsserver zur Verfügung stellen muss, sind im Rahmen der JEE-Spezifikation definiert
- Die genaue Umsetzung kann von Hersteller zu Hersteller differieren
  - Provider unterscheiden sich beispielsweise
    - im Grad der Modularität
    - in der Art der Konfiguration
    - in der Gestaltung von Administrations-Werkzeugen
- Auf Grund der gleichen technischen Vorbedingungen und den Zwängen der Spezifikation ergibt sich jedoch in der Praxis ein sehr ähnliches Lösungskonzept für zentrale Aufgaben
  - Request-Verarbeitung
  - Multithreading
  - Deployment



# Demo: Administrations-Oberfläche eines Applikationsservers

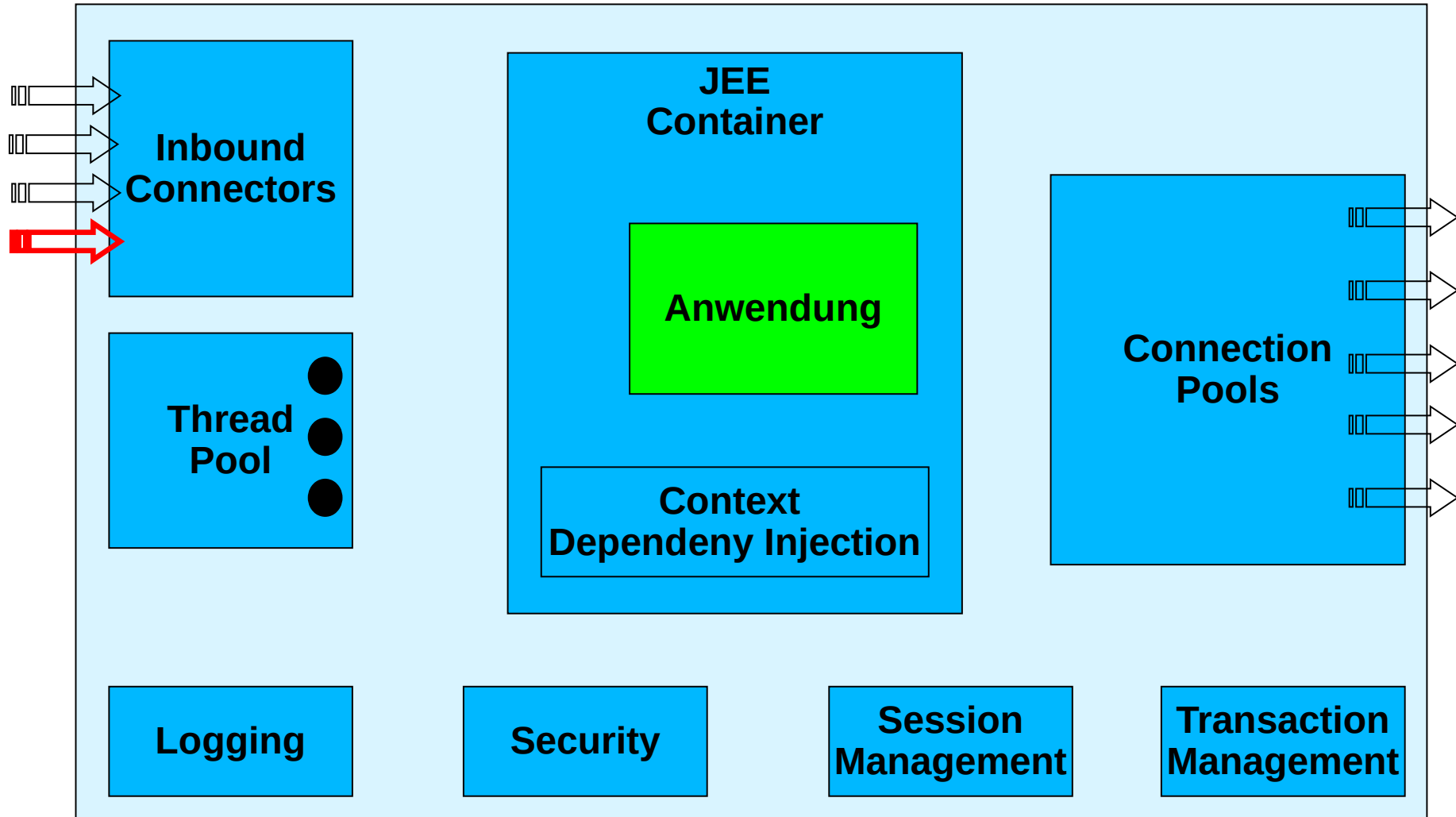


# Übung zu 2.1 – Arbeitsweise Applikationsserver

- Ziel:
  - Installation einer Applikationsserver + Applikation über das wildfly-maven-plugin
  - Überblick der Services über die Managementkonsole des Applikationsservers
- Schritte
  - Checken Sie das Projekt aus und gehen Sie in den Ordner examples/trainings
  - Starten Sie den Wildfly + Applikation wie in der README.md beschrieben
  - Finden Sie unter Deployments heraus, welche Anwendungen deployt sind
  - Finden Sie über Configuration die benutzten Ports heraus
  - Finden Sie unter Runtime heraus, wieviel Rest-Endpoints zur Verfügung stehen
  - Nennen Sie einige Services, die Sie auf der Managementoberfläche finden und ihre BedeutungMaterial zum Training

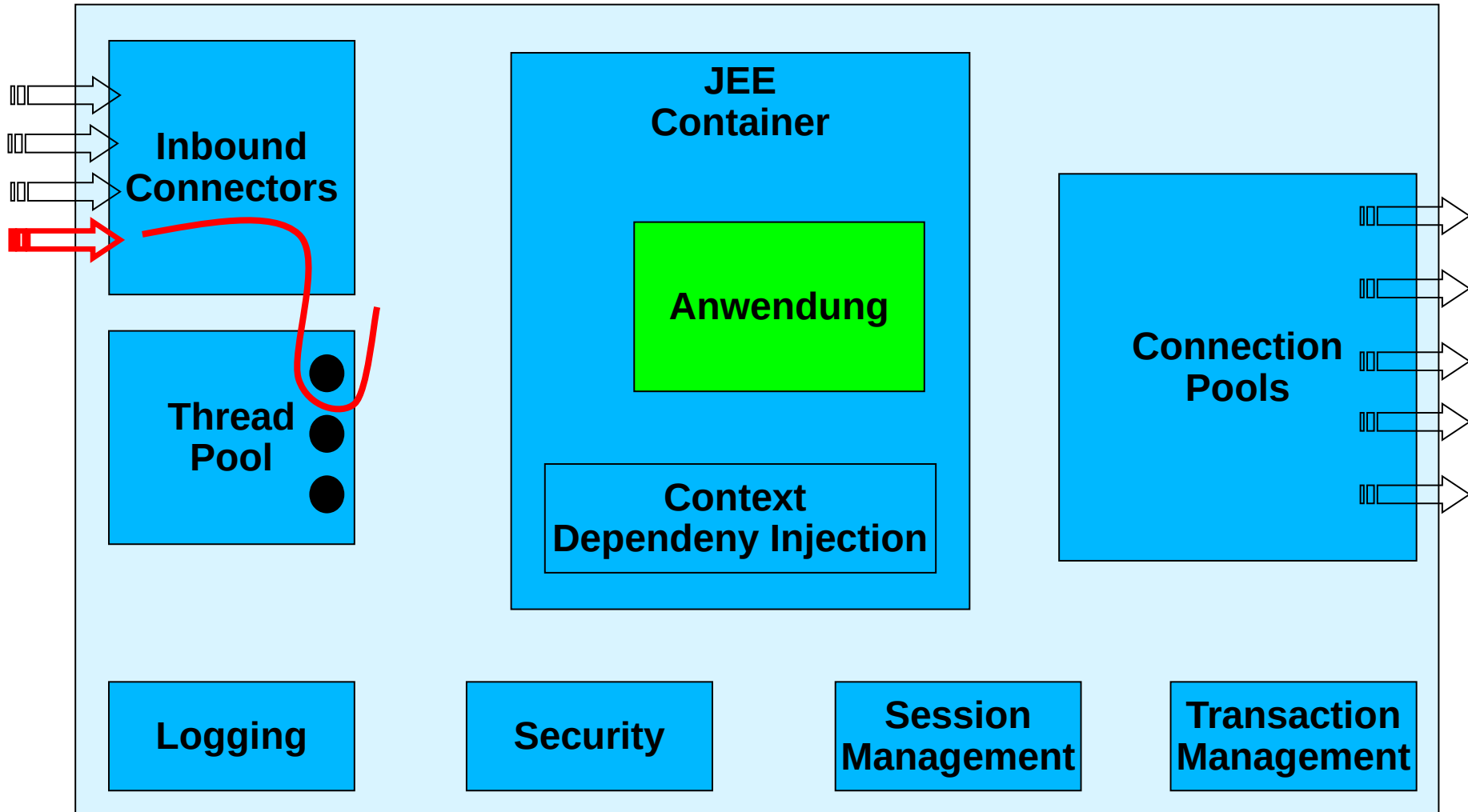
2.2

## **REQUEST VERARBEITUNG**

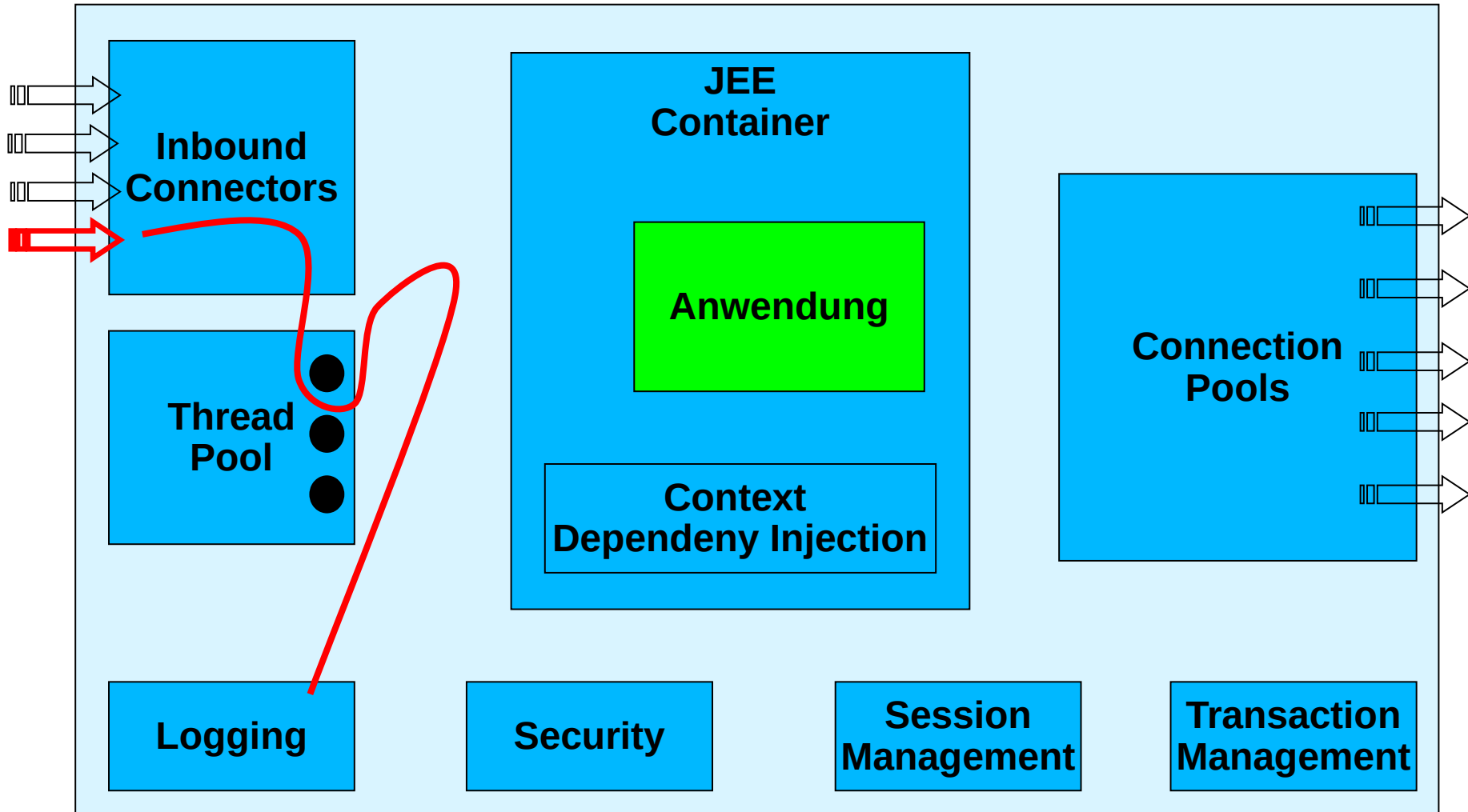




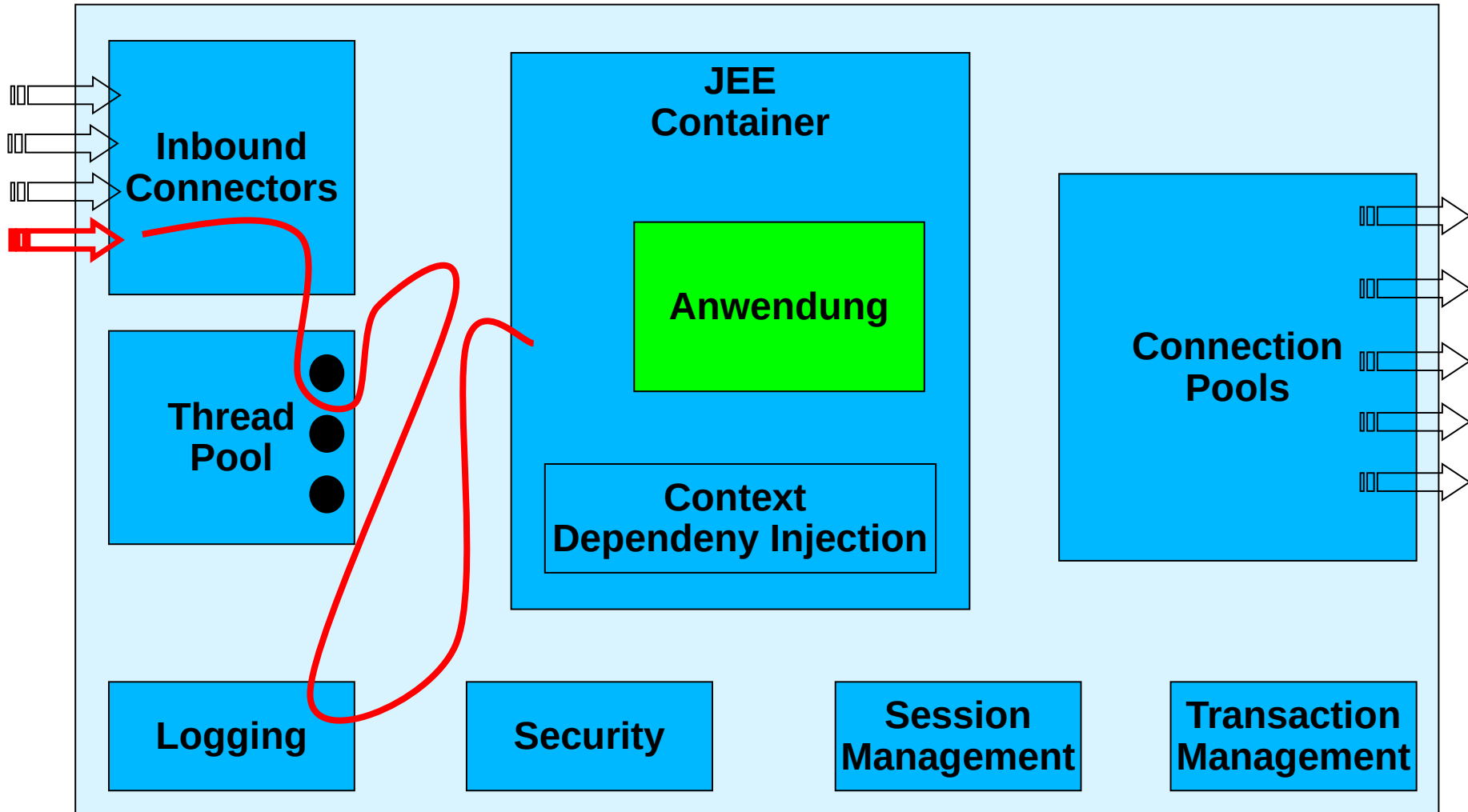
# Verarbeitung durch Thread aus Pool



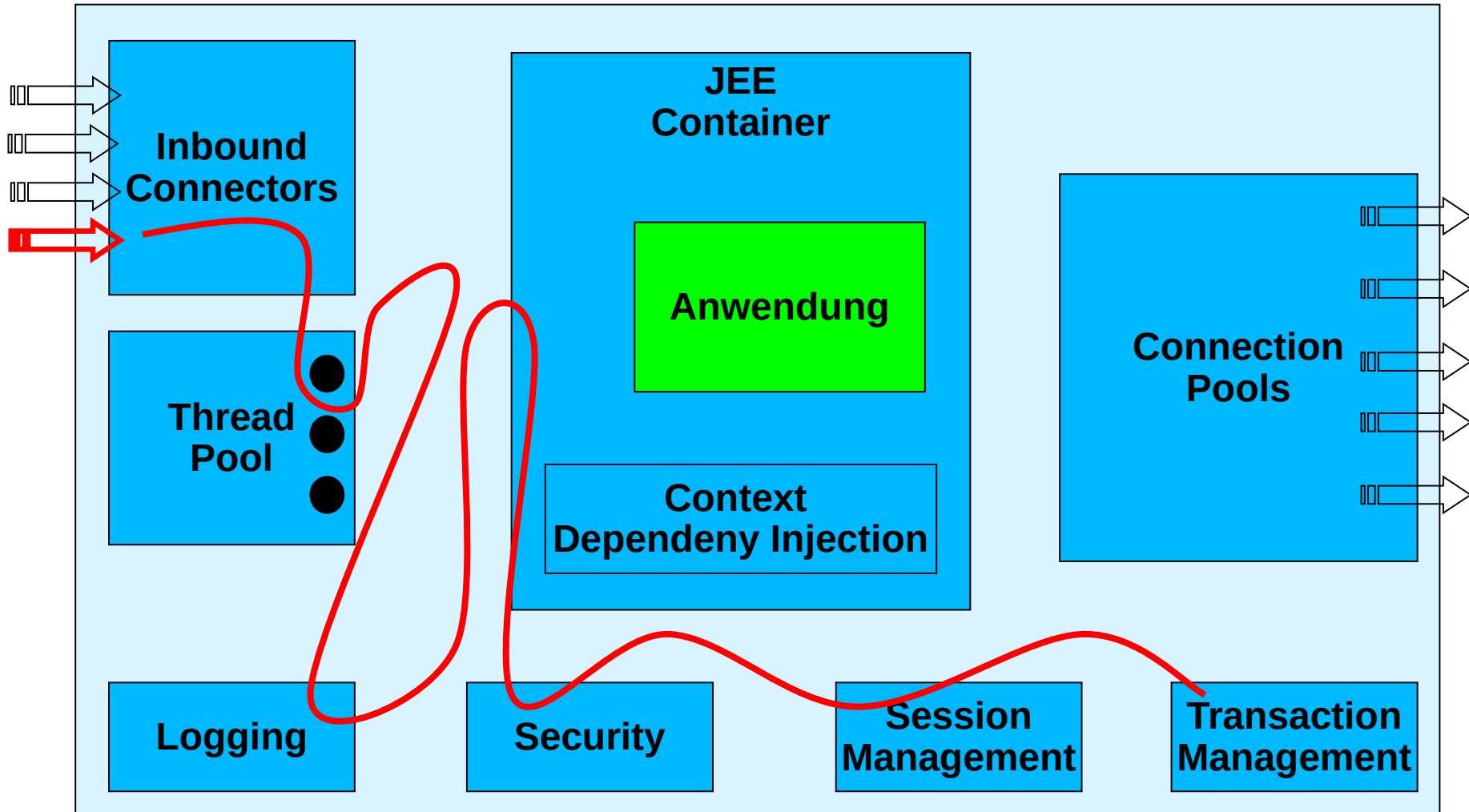
# Aufruf des zentralen Loggings

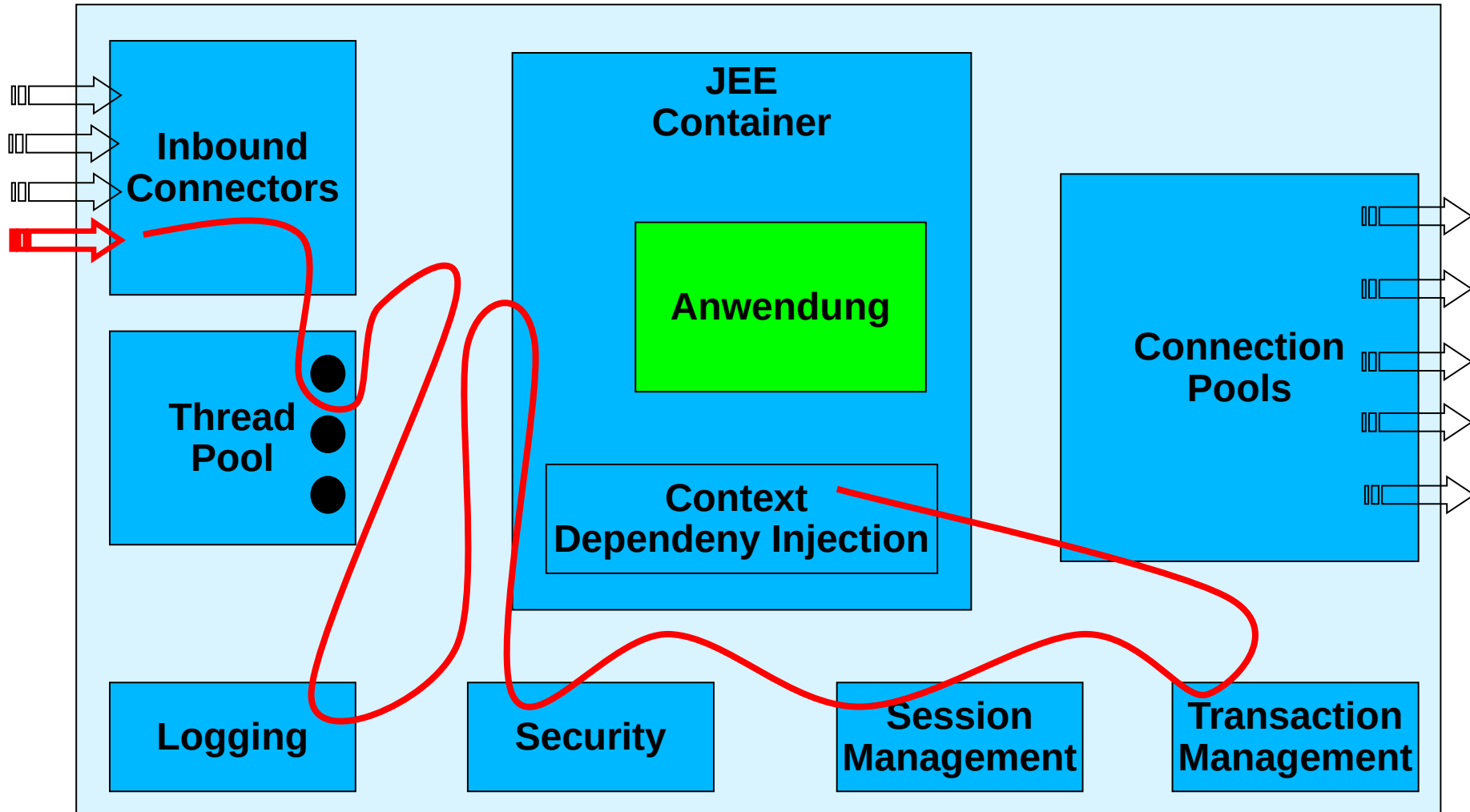


# Aufruf des JEE Containers

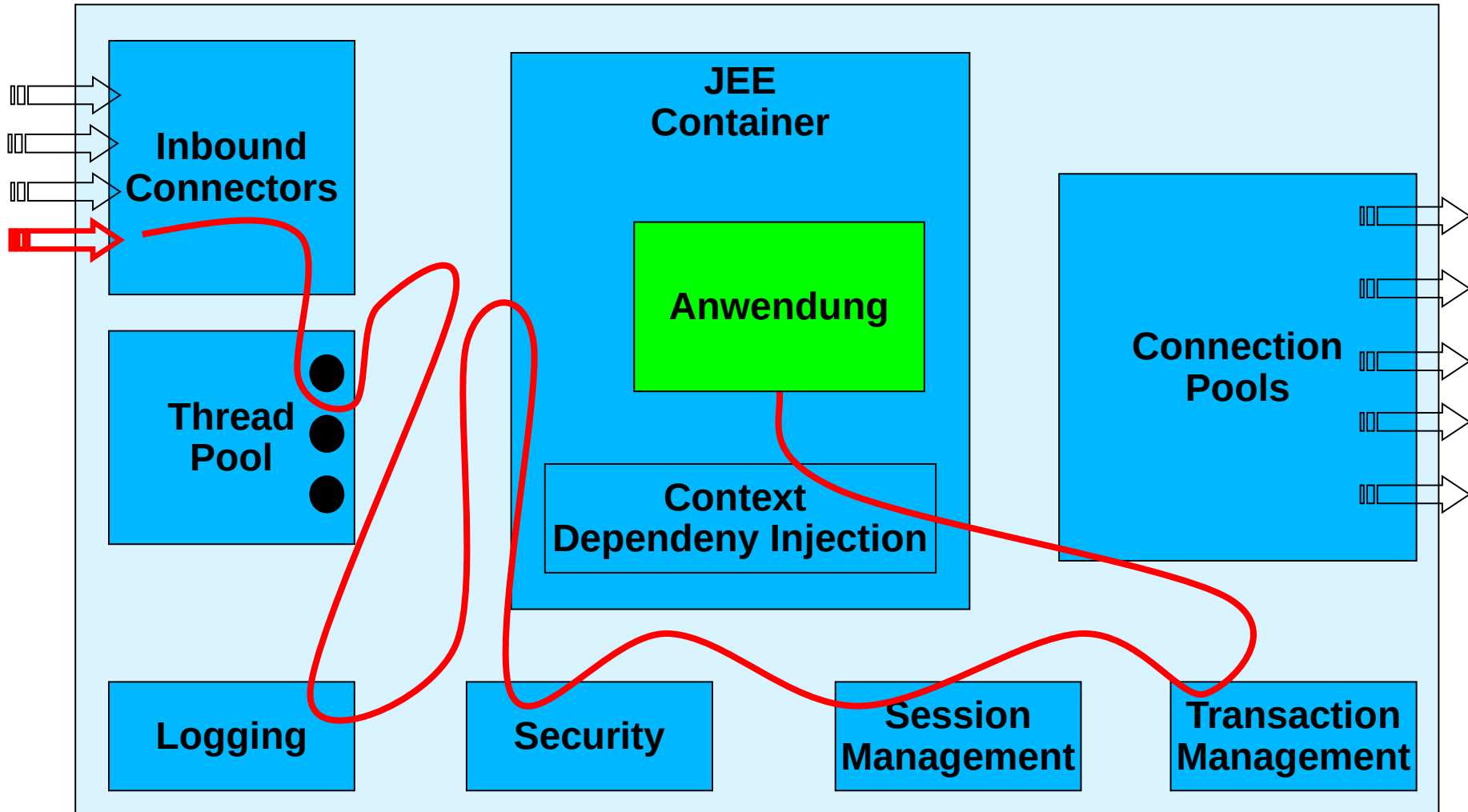


# Aufruf der zentralen Dienste

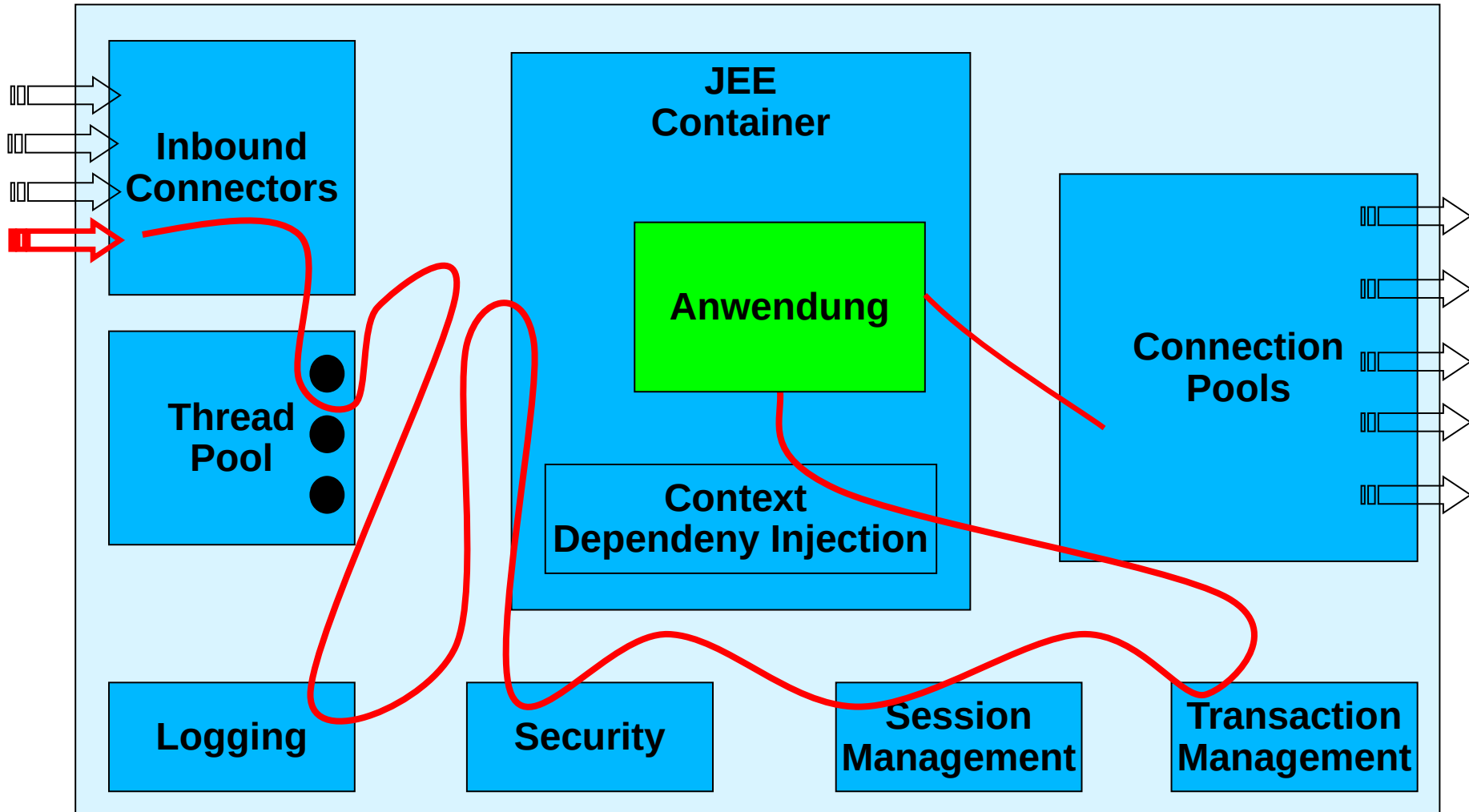




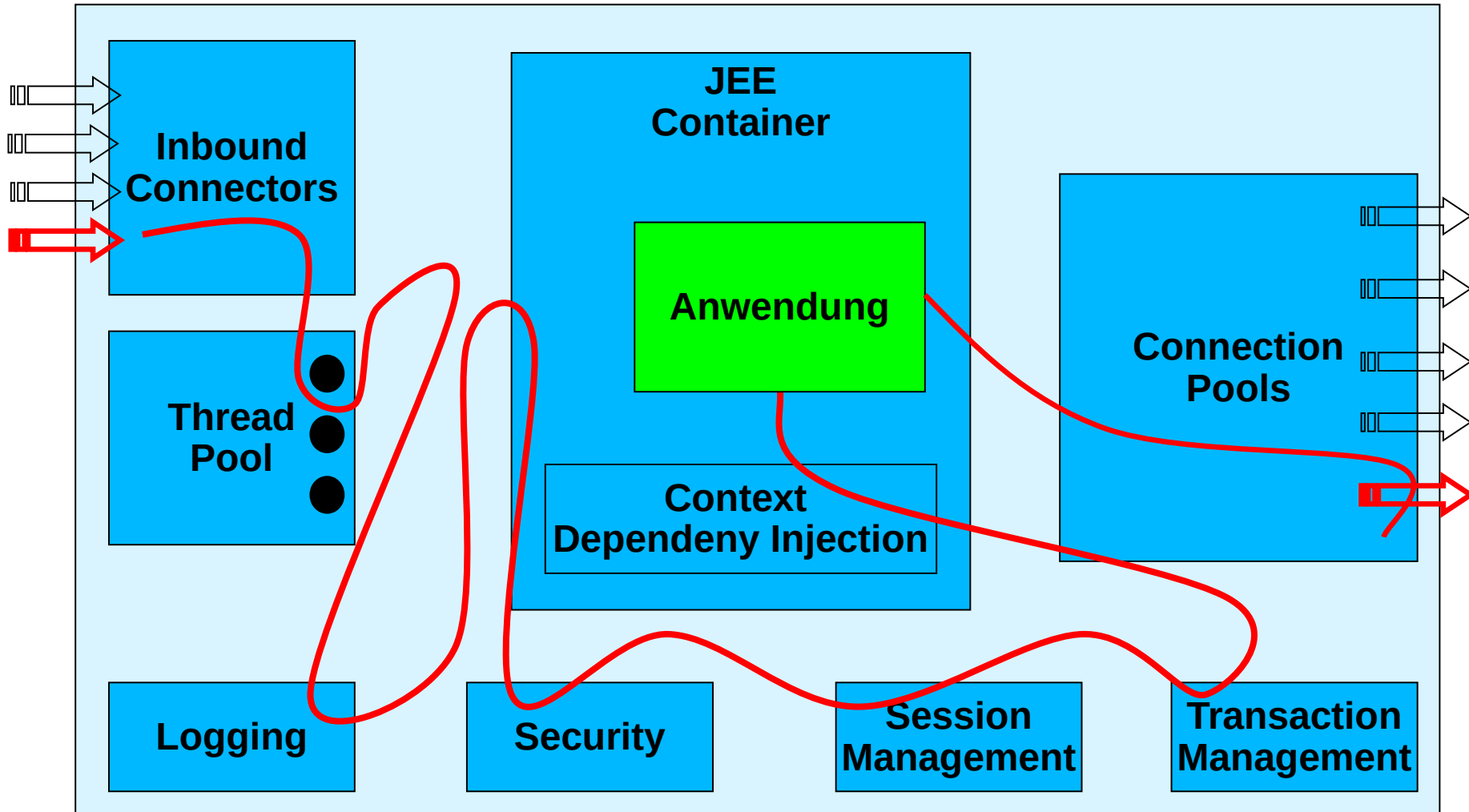
# Aufruf der Anwendungs-Logik



# Optional: Anforderung einer Connection

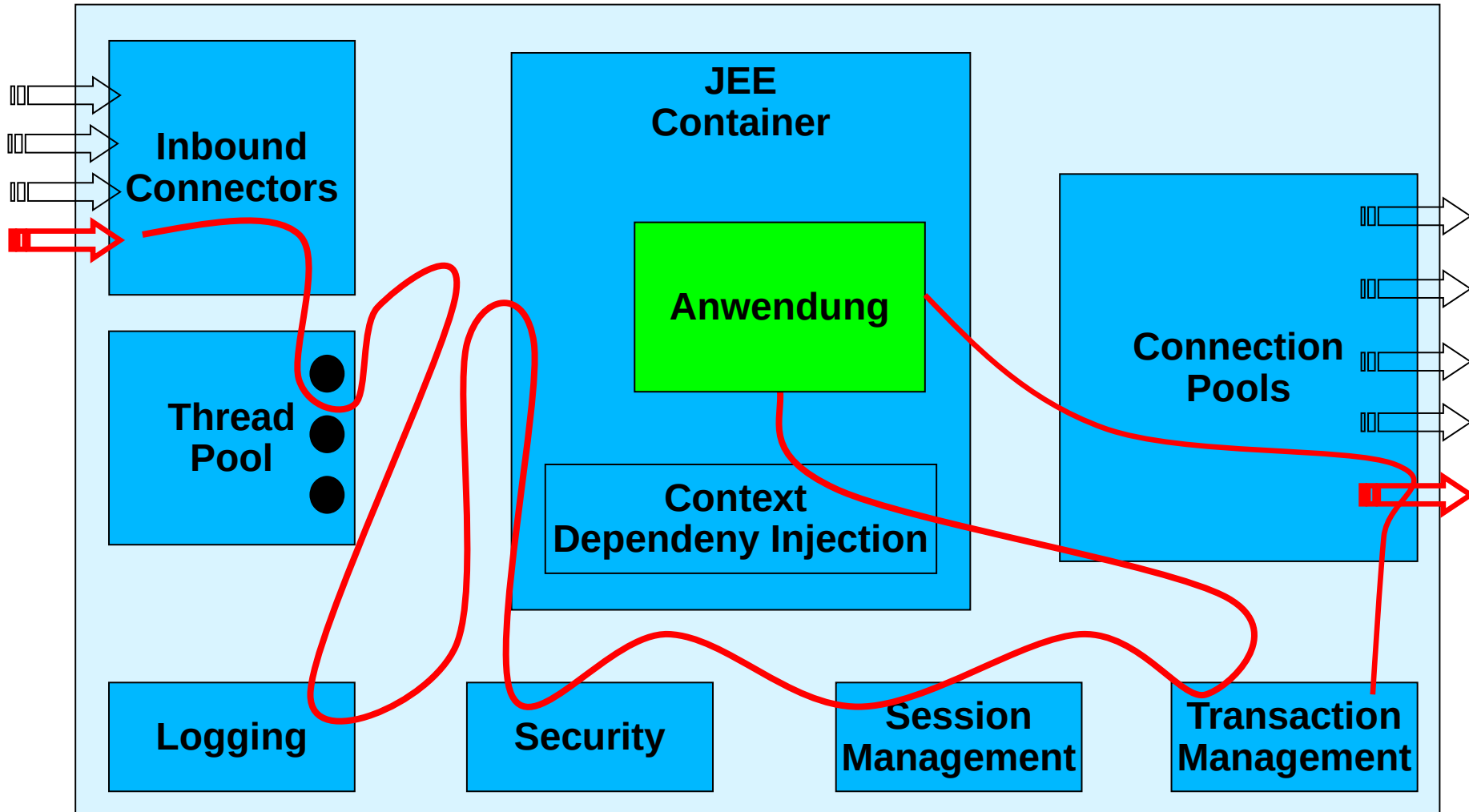


# Reservierung der Connection

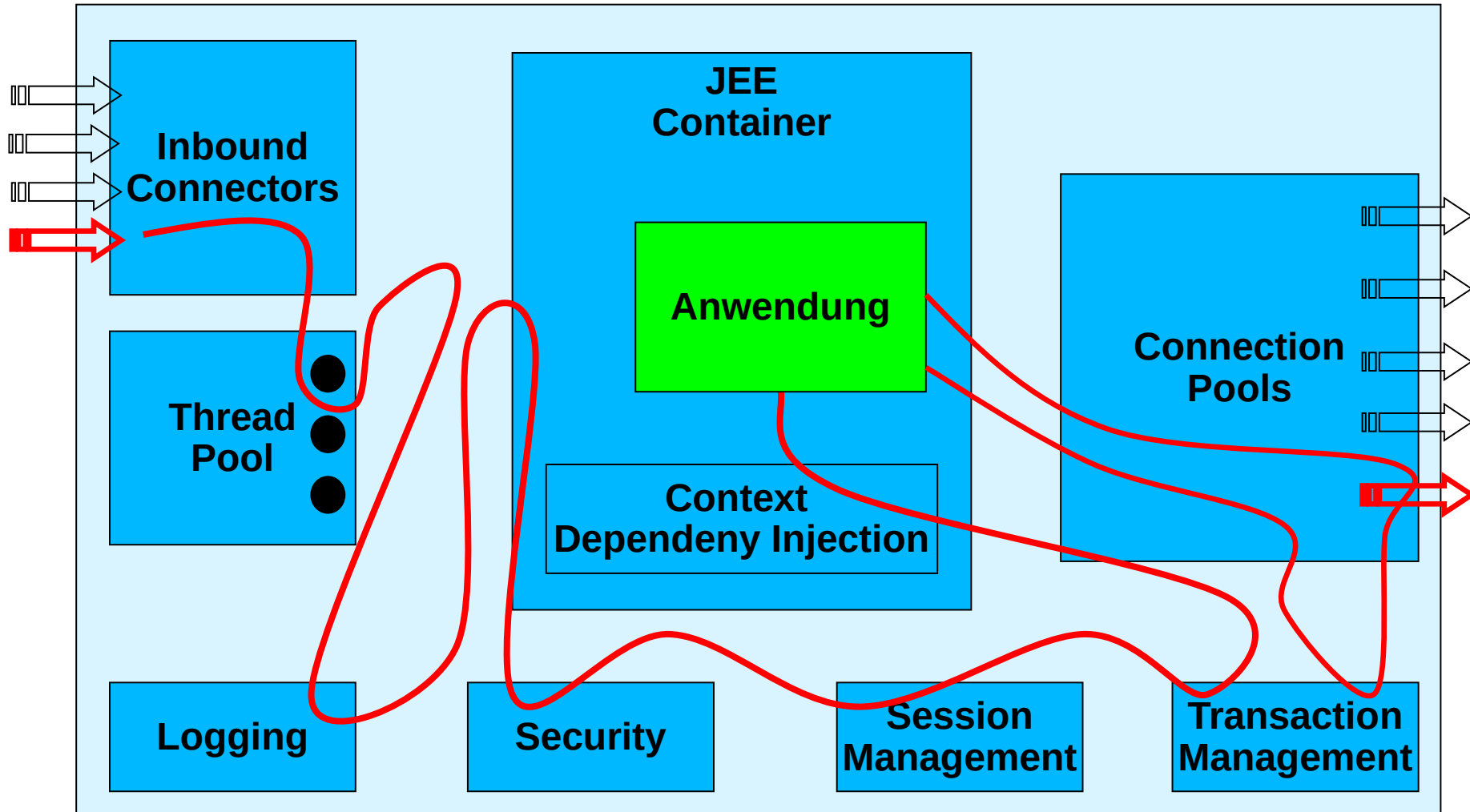




# Information des Transaction Managers



# Rückgabe der Connection an Anwendung





2.3

## DEPLOYMENT

- JEE Anwendungen sind gepackte Java-Archive
  - Optionale Zusatz-Informationen werden in XML-Deskriptoren hinzugefügt
- Anwendungs-Typen:
  - Anwendungen mit Enterprise JavaBeans
    - Endung .jar („Java Archive“)
  - Web-Applikationen
    - Endung .war („Web Archive“)
  - Spezielle Connector-Implementierungen
    - Endung .rar („Resource Archive“)
  - Gesamt-Anwendungen
    - Endung .ear („Enterprise Archive“)
    - Darin möglich sind Web-Archive, EJB-Archive und Resource Archive
- Die JEE 7 präferenziert als Packaging war-Dateien

- Das Deployment kann durch verschiedene Verfahren initiiert werden:
  - Auto-Deployment
    - Kopieren des Archivs in ein vom Hersteller definiertes Verzeichnis
  - Deployment-Tools, z.B. eine Web-Oberfläche
    - Wizard-gesteuerter File-Upload
  - Scripting
    - Hersteller-abhängige Skripte, die direkt mit dem laufenden Applikationsserver kommunizieren
  - Programmatisch
    - JEE definiert ein Deployment-API für Java-Anwendungen

- Das Deployment kann im laufenden Betrieb erfolgen
- Dabei können bereits installierte Anwendungen durch neue Versionen ersetzt werden
  - Interessant für Entwicklung und Test
  - Im Produktiv-Betrieb häufig nicht genutzt
- Hersteller implementieren teilweise eine ausgereifte Deployment-Prozedur
  - Parallele Bereitstellung verschiedener Versionen
  - Rollback-Mechanismus

# Deployment im Detail

- Wird eine neue Anwendung installiert wird eine komplexe Deployment-Sequenz durchgeführt:
  - Bereitstellung eines eigenen Klassenladers zur Kapselung verschiedener Anwendungen
  - Auslesen der im Archiv vorhandenen Klassen und Deskriptoren, um Meta-Informationen zu erhalten:
    - Transaktionsverhalten
    - Security
    - Anbindung an Inbound Connectors und damit Netzwerk-Protokolle
    - Abhängigkeiten und benötigte Ressourcen
  - Erzeugung des Containers und konkrete Anbindung an die Dienste des Applikationsservers
  - Instanziierung der Anwendungsklassen und Auflösen der Abhängigkeiten



# Übung zu 2.3 – Deployment

- Ziel:
  - Erste Schritte mit dem Wildfly Server, Durchspielen verschiedener Deployment-Arten.
- Schritte:
  - Über das wildfly-maven-plugin haben Sie im vorherigen Schritt bereits ein Wildfly Server heruntergeladen. Nun wollen wir diesen Standalone starten.
  - Kopieren Sie sich die den Ordner server im target-Verzeichnis.
  - Starten sie den Server mit `./bin/standalone.sh` bzw. `bin/standalone.bat`
  - Deployen Sie die WAR-Datei aus Ihrem target-Verzeichnis manuell
  - Löschen Sie Ihr Deployment wieder und deployen Sie die Anwendung erneut, indem Sie sie in das Verzeichnis deployments des Servers legen

2.4

## **ADMINISTRATION**

- Die Dienste des Applikationsservers müssen konfiguriert werden
  - Verzeichnisse
  - Pool-Größen
  - Timeouts
  - ...
- Konfiguration erfolgt über
  - Editieren von Konfigurationsdateien
  - Aufruf von Skripten
  - Benutzung einer Web-basierten Administrations-Oberfläche

# Übung zu 2.4 – Administration

- Ziel:
  - Administration über UI und standalone.xml
- Schritte:
  - Legen Sie unter Configuration eine H2-Datenbank an.
  - Schauen Sie sich an, wie sich die Serverkonfiguration unter server/standalone/configuration/standalone.xml geändert hat
  - Wieviele Datenbank Connection stehen im Connection-Pool zur Verfügung?

2.5

## ÜBERWACHUNG

- Betriebssystem
  - Standard-Werkzeuge der System-Überwachung
    - CPU
    - Speicher
    - Verfügbarkeit
- Java Virtual Machine
  - Java Management Extension (JMX)
    - Threads
    - Interne Speicherorganisation und Garbage Collection
    - Geladene Klassen

- Logging
  - Editor für Log-Dateien
    - Allgemeine Informationsausgaben
    - Fehlermeldungen
    - Detaillierte Debug-Ausgaben für Entwicklung und Test
- JEE Metriken
  - Zugriff über JMX oder Administrationswerkzeuge
    - Zugriffszeiten
    - Pool-Größen
    - Timeouts

- Der Applikationsserver schreibt eine Log-Datei
  - Enthält sämtliche Fehler-Meldungen sowie Status-Informationen
  - Gruppierung in einzelne „Log Level“



- Sämtliche relevanten System-Parameter werden überwacht
  - Pool-Größen
  - Speicher-Verbrauch
- Ebenso Zugriffszeiten auf die Anwendung

# Kapitel 2 - Wissenscheck

---

- Welche Arten des Deployments gibt es?
- Wie erreicht man die Administrationsoberfläche vom Wildfly (URL, Port)?
- Nennen Sie 2 Einstellungsmöglichkeiten in der Administrationsoberfläche
- Was muss man machen, um eine Postgres-Datenbank verwenden zu können?