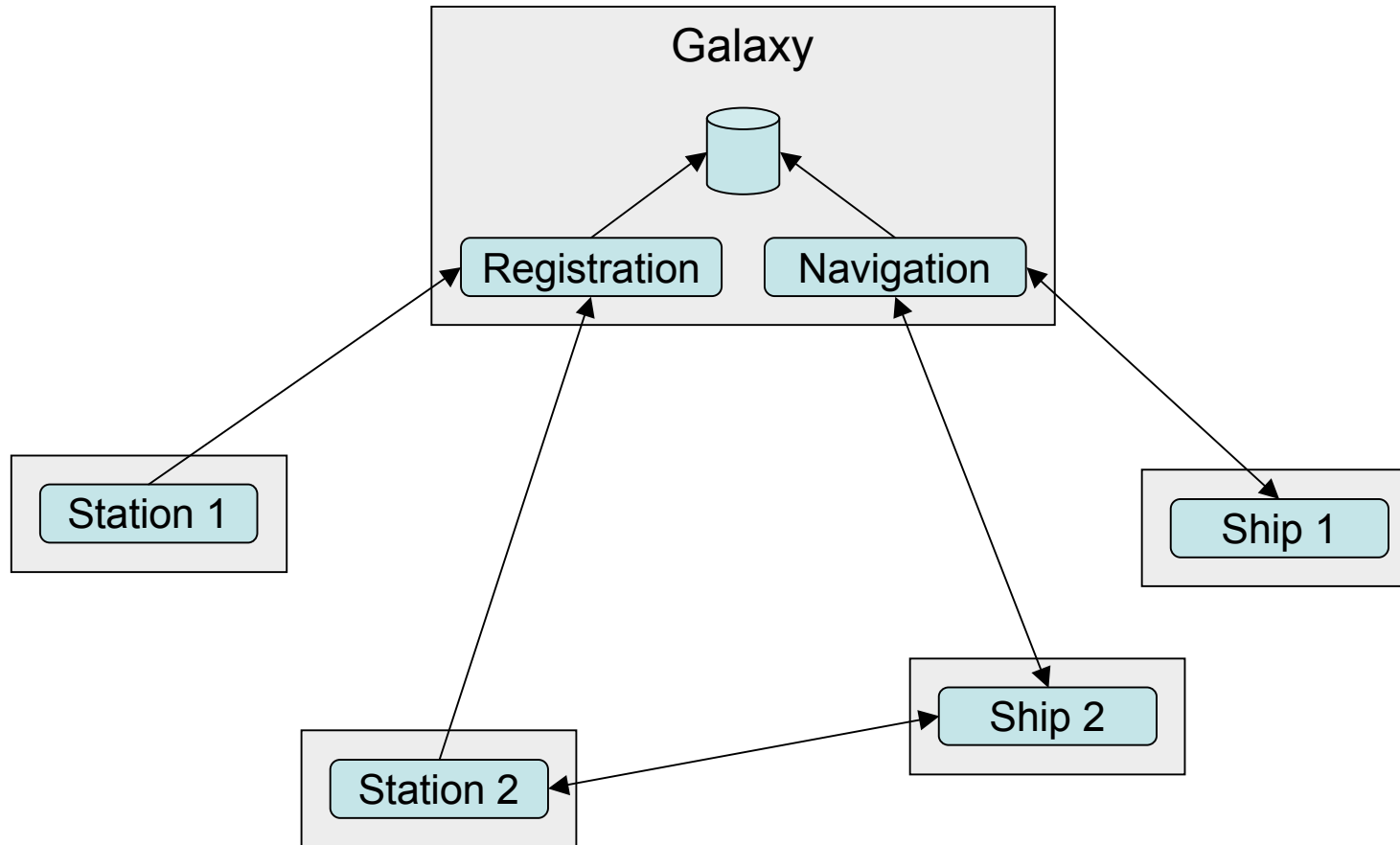


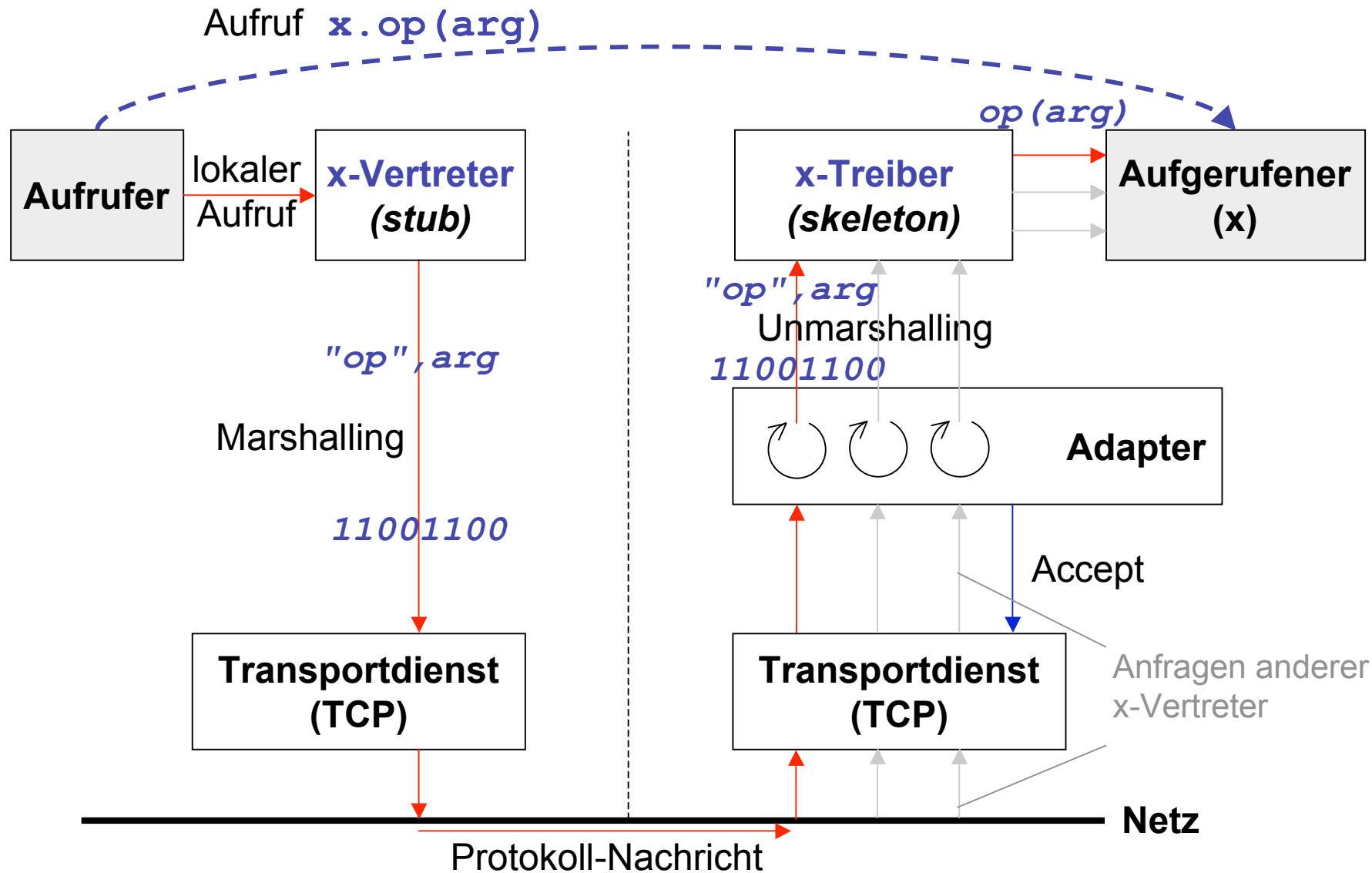
## **7. Aufgabenblatt**

**Fernaufrufsystem für Übung 6 selbst implementieren**

# Galactic Traders



# Fernaufrufe



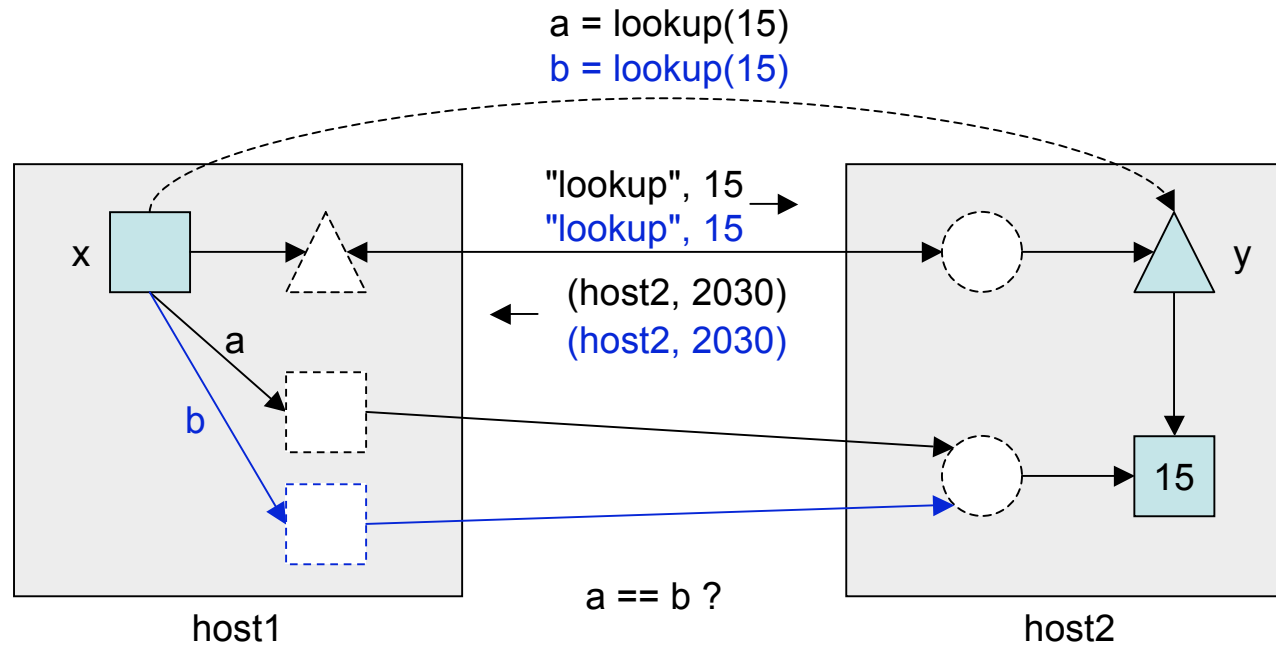
# Aufgabe

- **Stubs und Skeletons für Ship, Station, GalaxyRegistration, GalaxyNavigation (kein generisches System, keine Reflection)**
- **ein Adapter + Skeleton pro fernaufrufbarem Objekt, identifiziert über <host, port>**
- **Marshalling von Fernverweisen und Verbundtypen über Lese- und Schreiboperationen von Data [Input | Output] Stream (nicht Object- / kein Serializable) (flush() nicht vergessen)**
- **korrekte Identitätsprüfung**

## Objekte ↔ Fernverweise

- **Serialisierung von Parametern**
  - **primitiver Datentyp: siehe** `DataOutputStream`
  - **Verbundtyp: rekursive Serialisierung**
  - **Stub: <host,port> senden**
  - **lokales fernaufrufbares (callback-) Objekt**
    - **zugehöriges Skeleton finden, ggf. erzeugen**
    - **dessen <host,port> senden**

# Deserialisierung von Parametern



- primitive Typen/Verbundtypen: siehe Serialisierung
  - <host,port>
    - zugehörigen Stub finden, ggf. erzeugen
    - diesen übergeben
- ⇒ prozessweite Abbildung Objekte ↔ Fernverweise (Cache)

# Quelltext-Organisation

- Stub-Klassen haben gleichen Namen wie Implementierungsklasse (-> Verteilungsabstraktion), z.B.:

traders

  ship

    Ship.java (Implementierung)

    Station.java (Stub)

  station

    Ship.java (Stub)

    Station.java (Implementierung)

  ...