

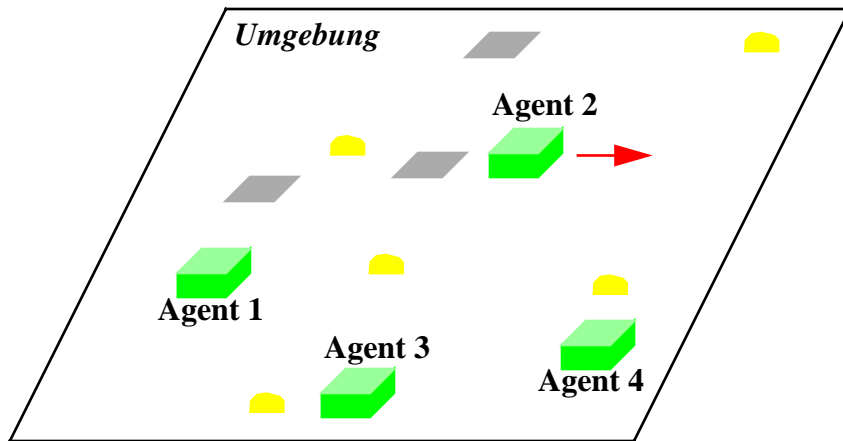
**Seminar“ Agentensysteme: Eine praktische Einführung“,  
WS 2000/2001**

## **Einführung in die WBS-Agenten (I)**

**2. Termin, 27.10.2000**

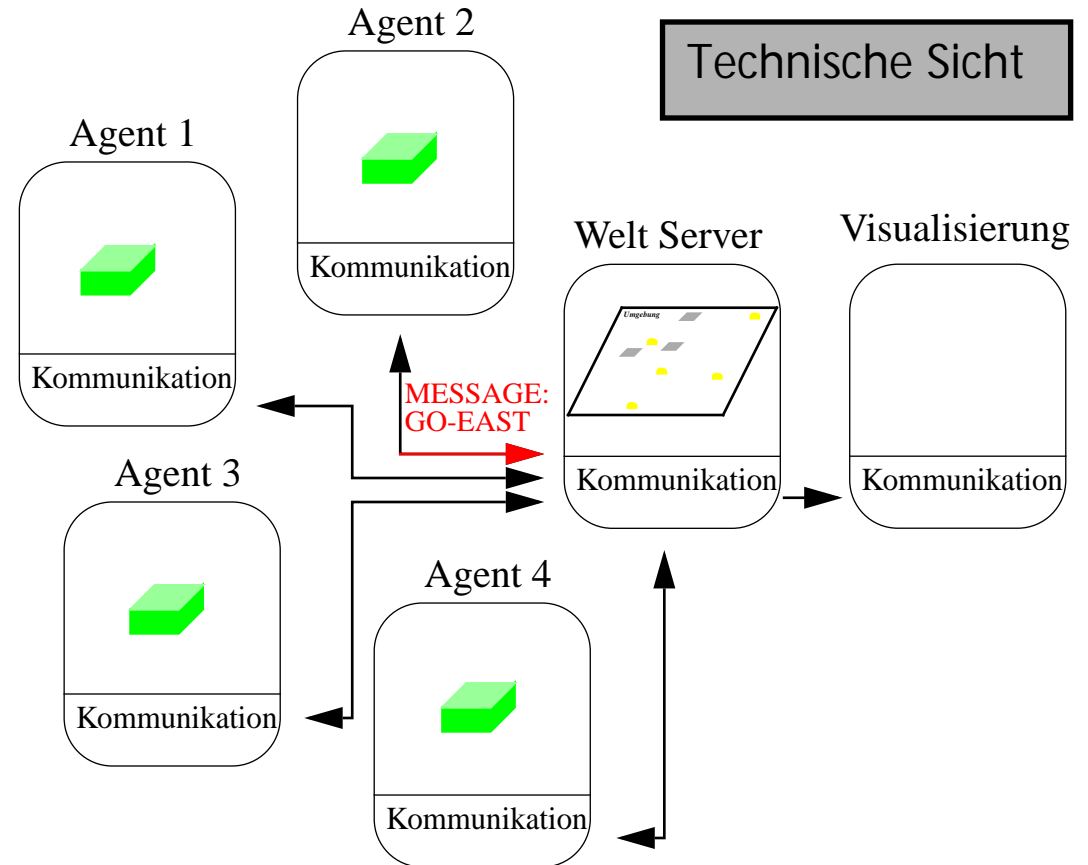
## Agenten

Blockwelt Sicht



Agenten (als „intelligente Individuen“) bewegen sich und interagieren in der Blockwelt.

Technische Sicht



Agenten (als „unabhängige Programme“) kommunizieren miteinander.

⇔ WBS-Agenten

## WBS-Agenten

Technische Sicht

- Programmierumgebung in der AG WBS für modulare, verteilte Systeme
- Softwareeinheiten mit einer Kommunikationsschnittstelle
- C++-Klassenbibliothek unter Solaris, Irix, (teilw.) Linux
- Ein System kann (quasi) aus beliebig vielen Agenten bestehen (Modularität, Redundanz).
- Alle Agenten können (theoretisch) mit in jedes System (Wiederverwendbarkeit).
- Die Agenten können beliebig im Internet verteilt sein (Verteiltheit).
- Agenten können zu jedem Zeitpunkt das System verlassen oder dem System hinzugefügt werden (Robustheit).

## Ein System aus WBS-Agenten

Technische Sicht

- Start durch „Maintainor“-Agenten (/vol/wbs/bin/CA\_Maintainor)
- Zentrale Konfigurierbarkeit durch zwei Dateien:

### agents.config

```
# SLOT Value
DEFAULTPATH /vol/wbs/share/Lehre/ws0001/AgentenSem/agenten/bin/


# SLOT Agentname Agentfile Hostname Display FlagString Options
# BEGIN_SLOTLIST AGENT
Monitor AgentstateMonitor_sol2 - - +
Viewer Viewer bootes - -
WorldServer WorldServer lynx - -
Agent1 TestAgent - - -
Agent2 TestAgent - - -
Agent3 TestAgent - - -
# END_SLOTLIST
```

### defaults.config

```
# SLOT Value
FieldSize 20
WorldSize 100
VisDistance 6
World-KB
/vol/wbs/share/Lehre/ws0001/AgentenSem/
commonbin/Welt.dat
cody-system
/vol/wbs/share/Lehre/ws0001/AgentenSem/
agenten/bin/
```

- Kontrolle und Statusanzeige über Monitor-Agent:

Technische Sicht

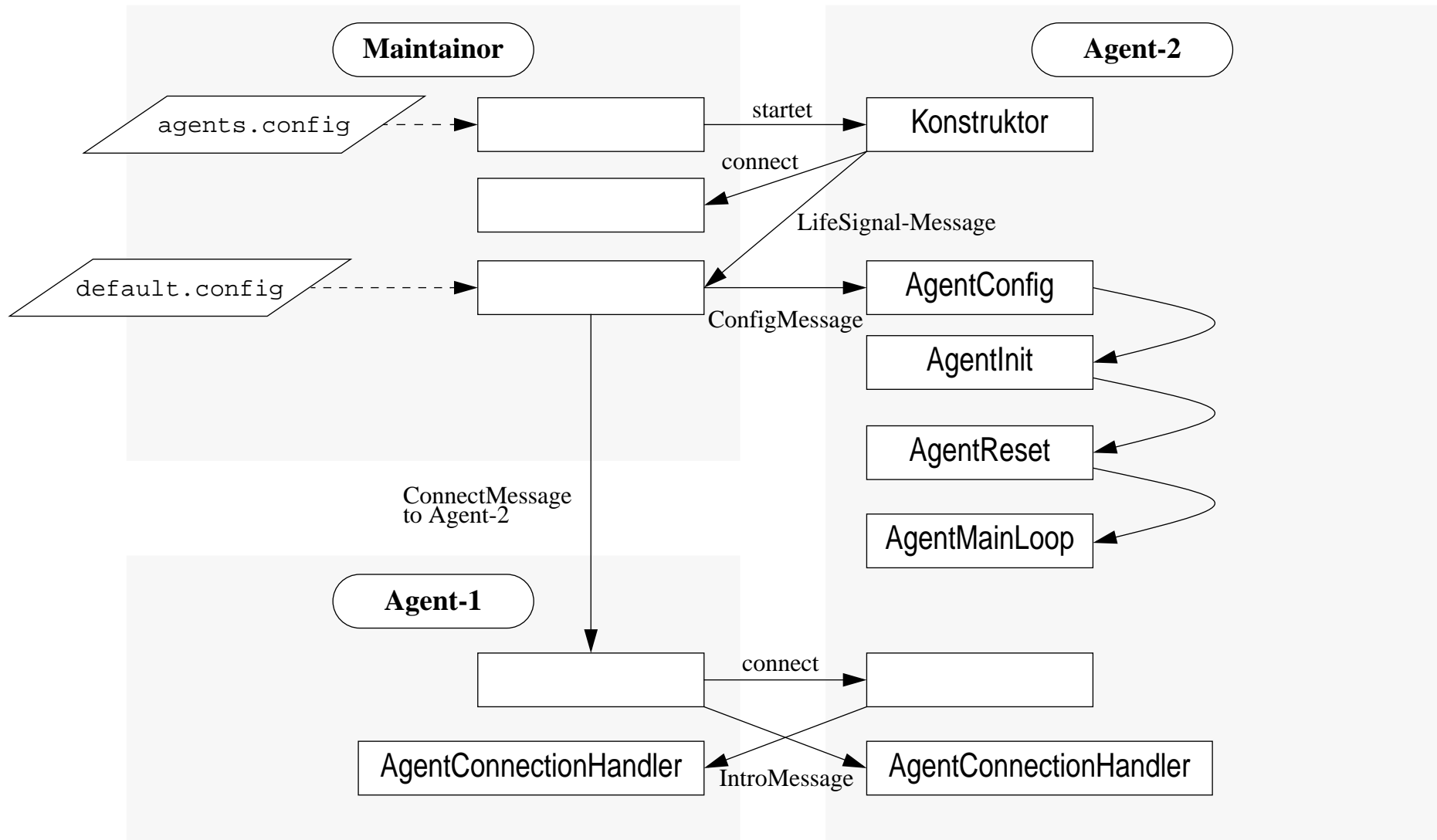


The screenshot shows a window titled "Monitor" with two buttons: "Ping" and "Exit System". Below the buttons is a table with three columns: "Alive", "Dead", and "Agent". The table lists the status of various agents in the system.

Alive	Dead	Agent
◆	◇	Monitor
◆	◇	Viewer
◆	◇	WorldServer
◆	◇	Agent1
◆	◇	Agent2
◇	◆	Agent3
◇	◆	Agent4
◇	◆	Agent5
◇	◆	Agent6
◇	◆	Agent7
◇	◆	Agent8
◇	◆	Agent9
◇	◆	Agent10

## Startvorgang eines Agenten

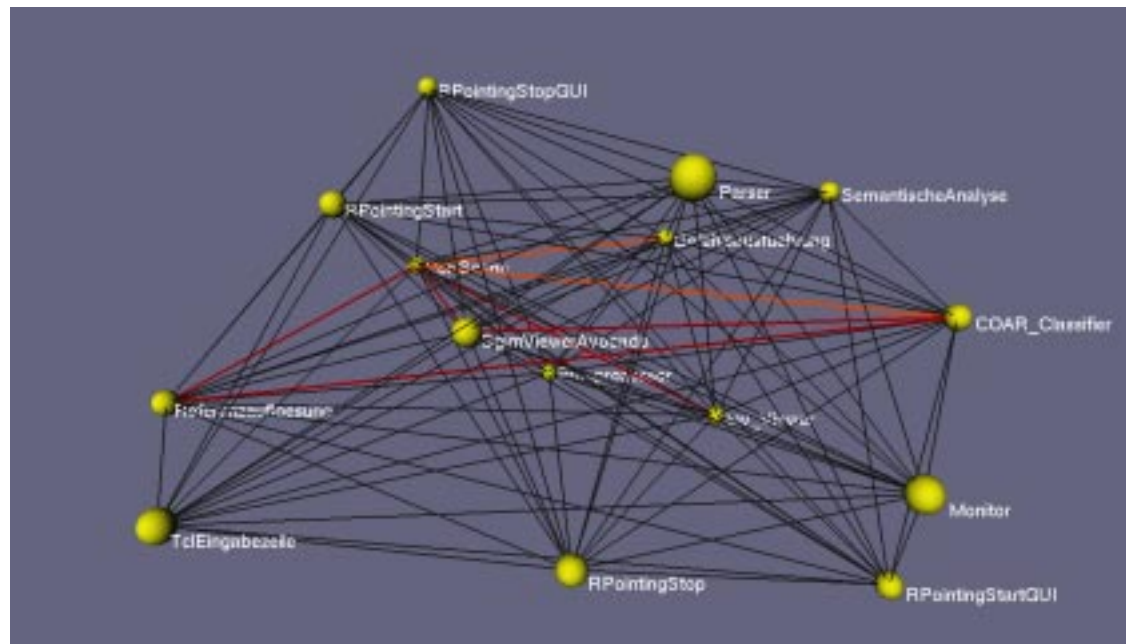
Technische Sicht



## Kommunikation

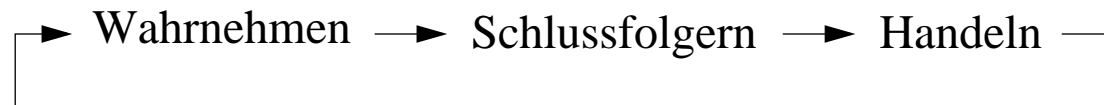
Technische Sicht

- Jeder Agent „kennt“ jeden anderen mit Namen
- Zwischen je zwei Agenten gibt es einen eigenen Kommunikationskanal
- Agenten schicken, empfangen und verarbeiten „Messages“ (Nachrichten-Objekte)

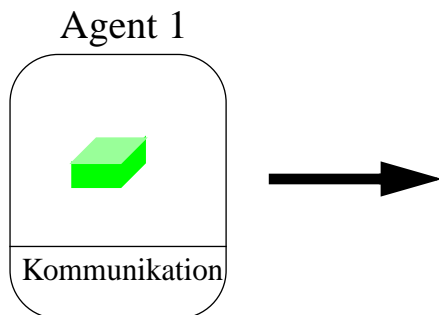


## Innenaufbau eines Agenten

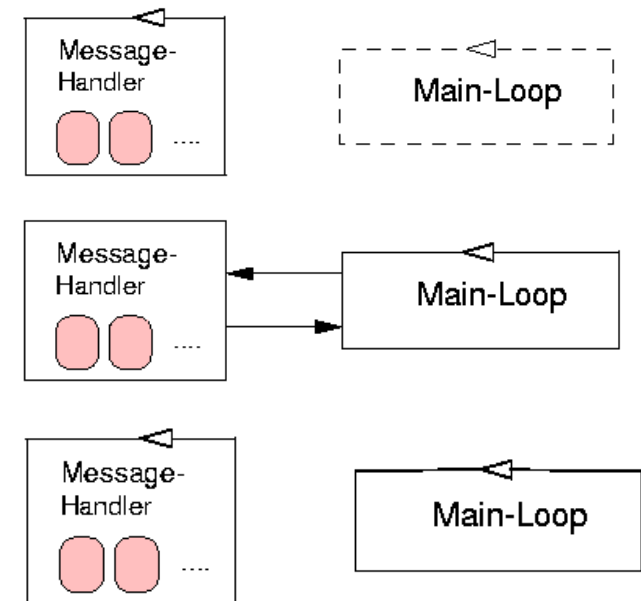
Blockwelt Sicht



Technische Sicht



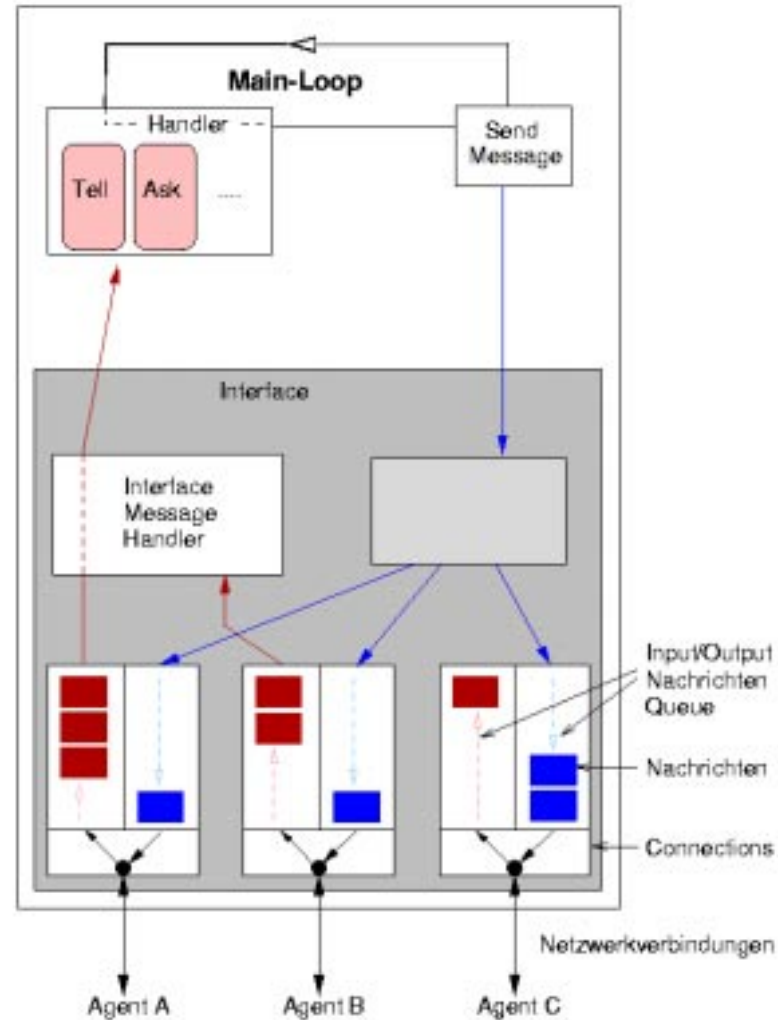
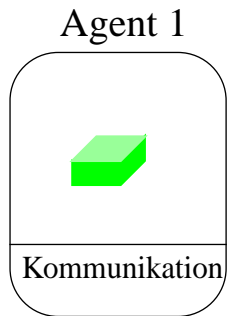
- „InactiveAgent“: rein nachrichtengetrieben
- „ActiveAgent“: manuelle Nachrichtenverarbeitung aus Main-Loop
- „ActiveThreadAgent“: Nebenläufige Nachrichtenverarbeitung und Main-Loop





## Innenaufbau eines Agenten

Technische Sicht

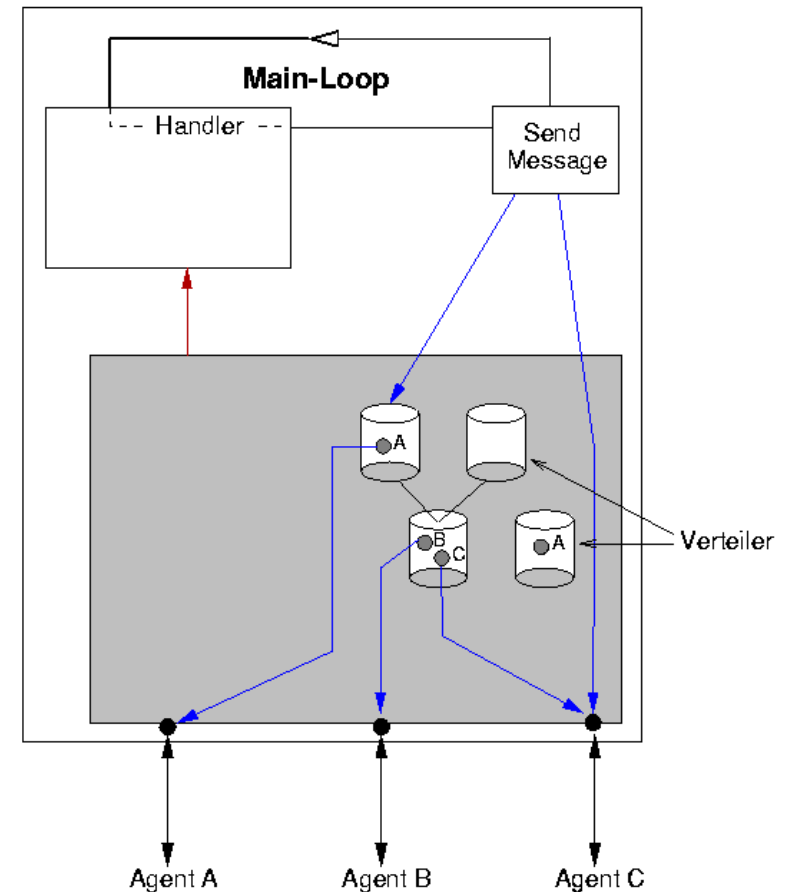


## Kommunikation

Wer kriegt welche Nachricht und wann und überhaupt?

⇒ „Verteiler“

- „Behälter“ von Kommunikationspartner
- Oft informationsbezogene Verteiler, z.B. „*World-Data*“
- Selbstständiges Erstellen der eigenen Verteiler
- Selbstständiges (`SubscribeMessage`) oder „erzwungenes“ Eintragen auf anderen Verteilern
- Ein Agent bekommt mit, wenn sich ein anderer Agent auf einem seiner Verteiler einträgt



## Kommunikation und Kooperation

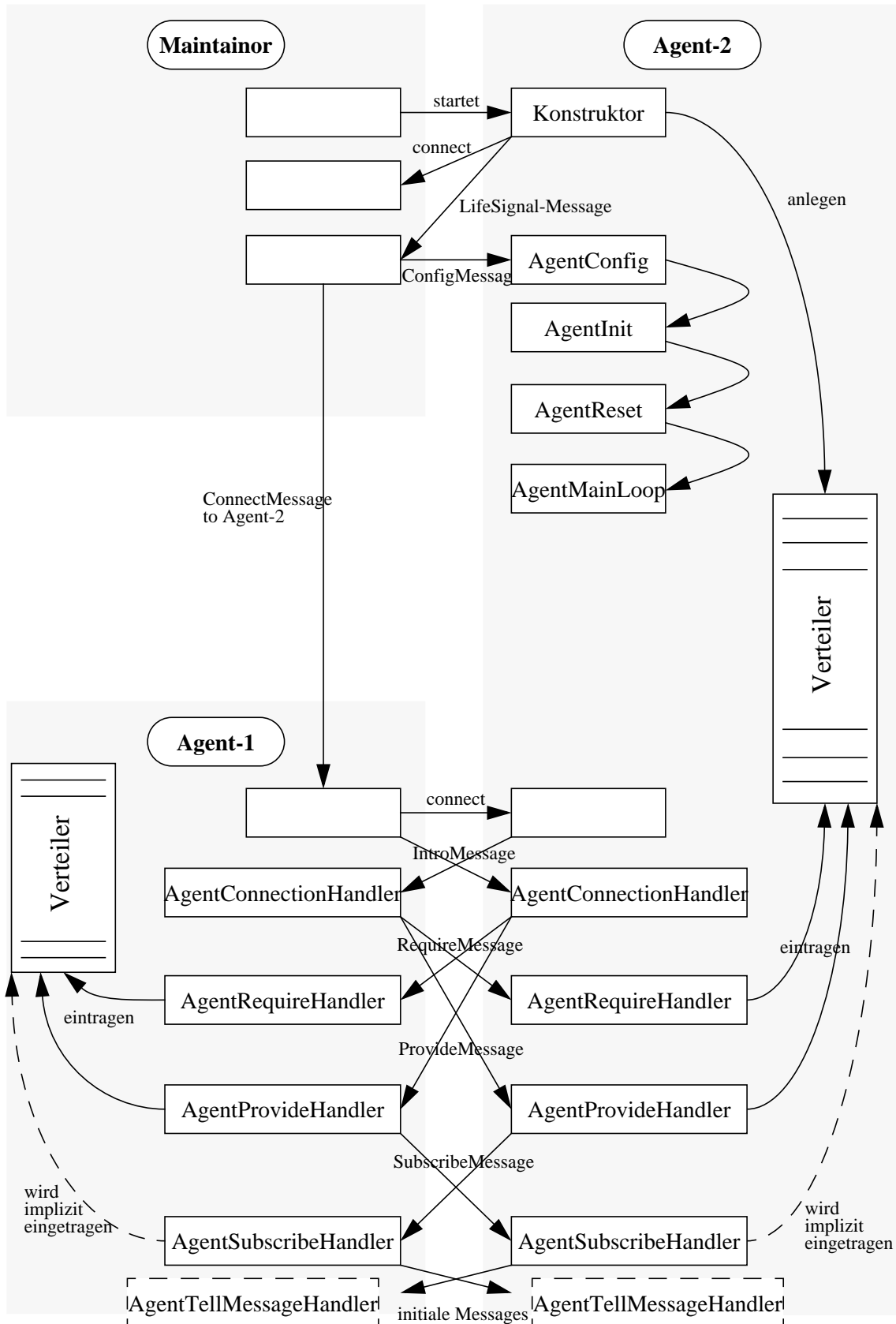
Auf welchem Verteiler muss ich mich eintragen, um die nötigen Informationen zu bekommen?

⇨ „Dienste“

- Beschreibt eine bestimmte Funktion bzw. Dienstleistung im Agentensystem, z.B. „*Aktuelle Informationen über die Blockwelt*“
- Gekennzeichnet durch einen (systemweit!) eindeutigen Namen, z.B. „*World-Data-Service*“
- Ein Agent kann Dienst(e)...
  - anbieten („*Ich biete...*“) ⇨ ProvideMessage
  - nachfragen („*Ich brauche...*“) ⇨ RequireMessage
- Dienste sind i.d.R. an Verteiler geknüpft: „*Ich biete aktuelle Informationen über die Welt auf meinem Verteiler World-Data*“  
⇨ ProvideMessage („World-Data-Service (World-Data)“)

## Das Kennenlernen...

Technische Sicht



## Seminarumgebung

- Verzeichnisstruktur: Alles unter `/vol/wbs/share/Lehre/ws0001/AgentenSem`
- Beispielsystem im Unterverzeichnis `.../commonbin/` ⇔ Kommando: „go“ (Link auf Maintainor)
- Erste Aufgabe: Eigenes Agentensystem einrichten (\*.config-Dateien kopieren und Maintainor starten)

## Organisatorisches

- Gruppe & Mailverteiler: `semagent`
- SGI-Rechner auf M4-112: `lynx`, `bootes`, `draco`, `pyxis`, `fornax`, `andromeda` (WorldServer und Viewer laufen nur hier!)

## Nächste Woche

- Einführung in die Blockwelt
- Wie baue ich meinen eigenen Blockwelt-Agenten? (⇔ C++ Handout lesen!)