Agentenmanagement

Kommunikation über Pfade
Waisenerkennung

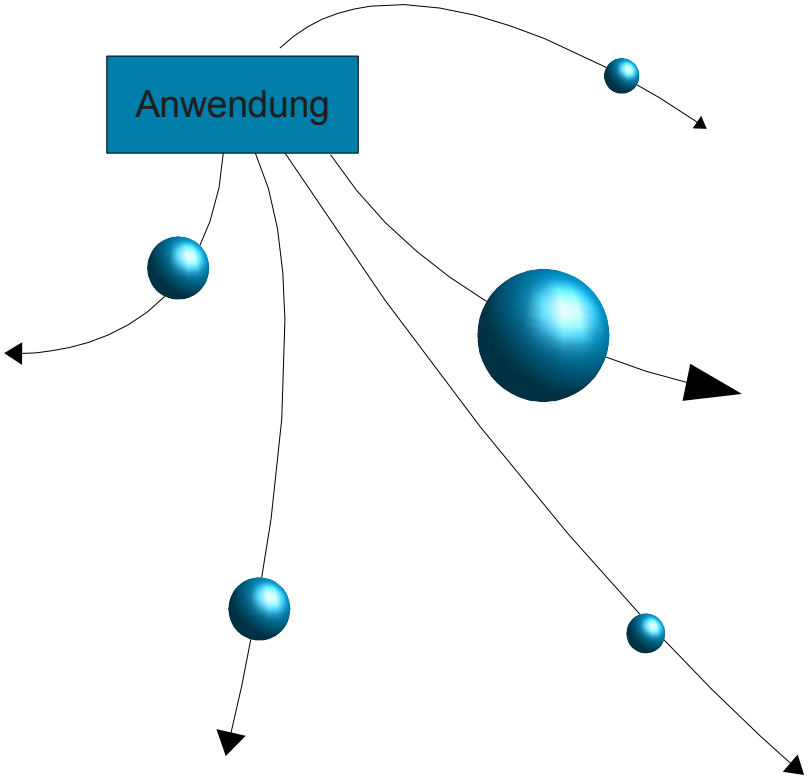
14.05.2007

Management von Agentenanwendungen

- Management
  - was zum Erhalt der Funktionalität beiträgt
  - was zur Stabilität des Systems beiträgt
  - was den Betrieb von Anwendungen erleichtert
- Infrastruktur oder Anwendung?
  - Anwendung: Spezifisch gestaltet, muss für jede Anwendung einzeln programmiert und gelernt werden
  - Infrastruktur: Allgemein gestaltet, muss nur einmal programmiert und gelernt werden

Auffinden von mobilen Agenten

- Anwendung benutzt mobile Agenten
  - verteilen sich im Netz
  - Anwendung hat Aufgabe erfüllt
  - Agenten müssen informiert werden
- Verschärftes Problem der verteilten Terminierung
  - Welchen Zustand hat eine verteilte Anwendung
  - Konsistente Sicht?



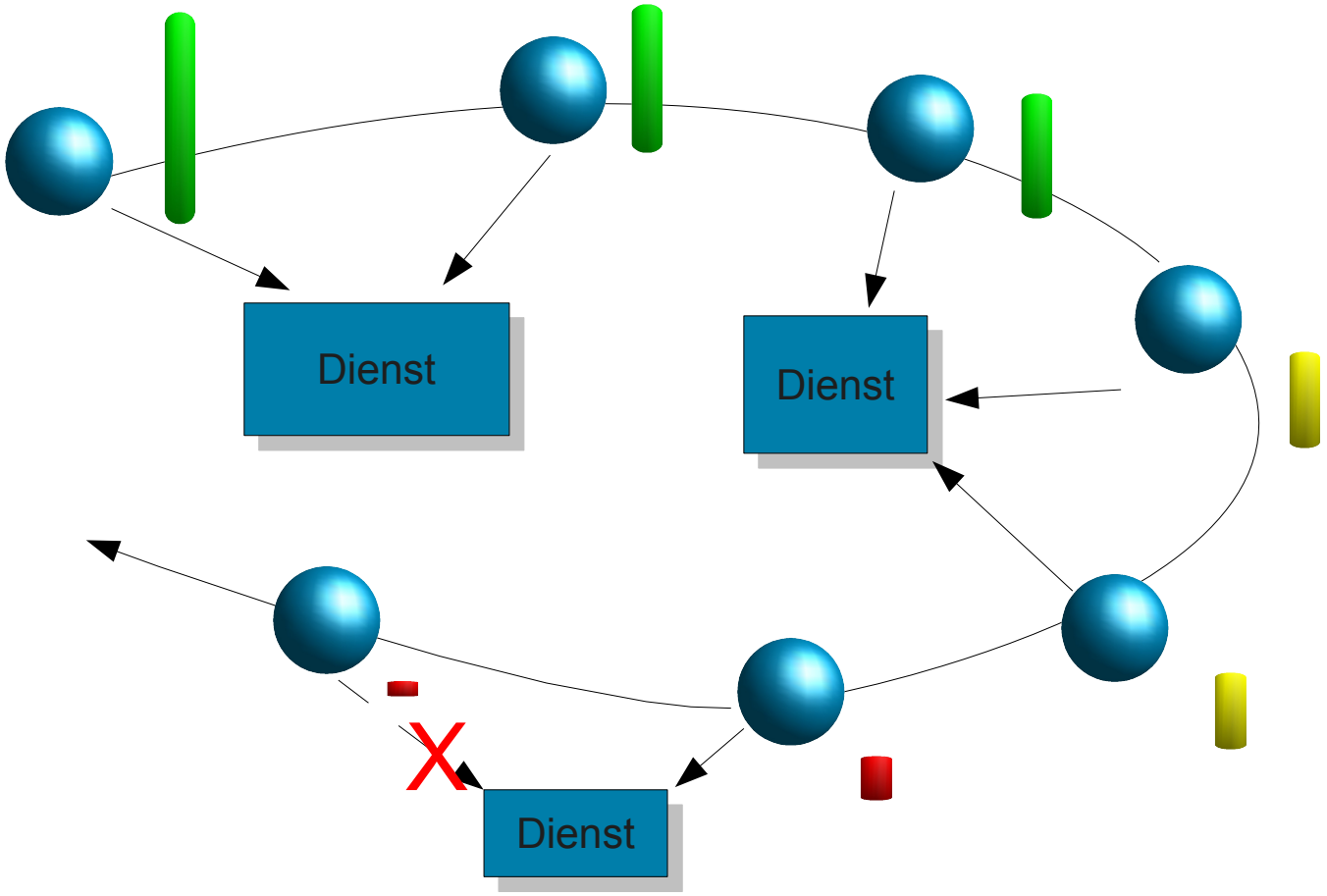
Auffinden von Agenten

- Ansatz: Waisenerkennung (Orphan detection)
  - bekannte Strategie, aber erweitert für MA
- Kombination mit Pfadsuche
  - Vertreter: Baumann, Rothermel (Uni Stuttgart)

Energiekonzept

- Agent nutzt Dienste und damit Ressourcen
- Sicherheitsprobleme
  - Unberechtigter Zugriff
  - Übermäßiger Verbrauch von Ressourcen (damit Blockieren des Dienstes, DoS)
- Abhilfe
  - Virtuelles Geld: Jede Dienstleistung kostet etwas
  - Energie: Jede Inanspruchnahme kostet Energie

Energiekonzept



Energiekonzept

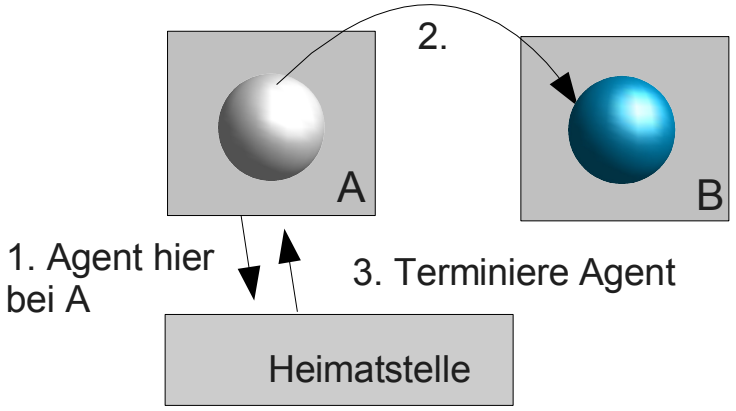
- Hauptnutzen des Konzepts
  - Agenten unschädlich, sobald ihre Energie verbraucht ist
  - Autorisation liegt außerhalb des Agenten: keine Möglichkeit, bei gegebenem Energiestand zu mogeln
- Probleme
  - Wer speichert die Energie?
    - nur der Agent: Betrugsgefahr
    - externes zentrales Konto: Kommunikationsaufwand
    - lokales Konto (Pro-Stellen-Guthaben): Missbrauch durch Stellenwechsel
  - Wer ergänzt die Energie?
    - Auffrischung für dauerhaft laufende Anwendungen notwendig

Energiekonzept

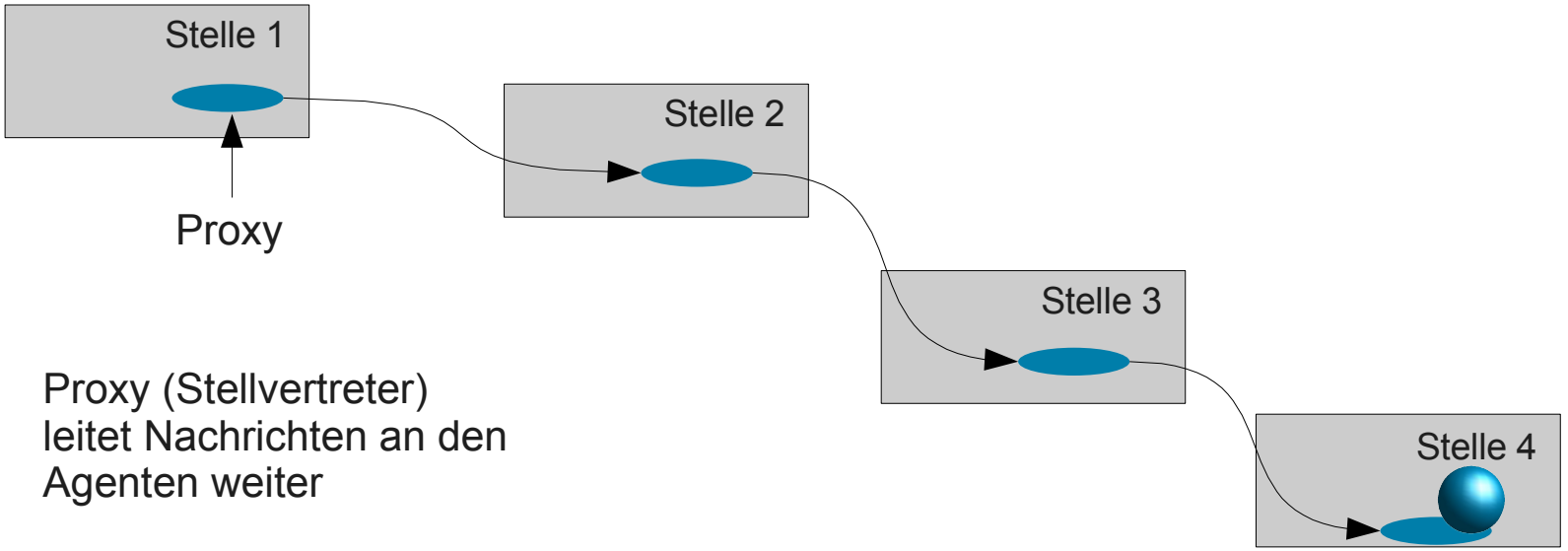
- Beenden der Auffrischung
  - Agent „verhungert“
  - Stelle kann ihn terminieren
  - kann auch passieren, wenn Anwendung abstürzt (daher: Agent „verwaist“)
- Abhängig von Kommunikationskanal zur Anwendung
  - Anwendung läuft noch, aber keine Verbindung: Agent wird fälschlich als verwaist erkannt
  - Umgekehrt wird jede Waise sicher erkannt

Pfadkonzept

- Anwendung möchte alle zugehörigen Agenten stoppen
  - Aufgabe: Finden und Terminieren von Agenten
  - Annahmen
    - System ermöglicht das Senden von Nachrichten an entfernte Stellen (andernfalls spezieller Agent zu verwenden)
    - Terminierung möglich durch Schicken einer Terminierungsnachricht
- Mobilität erschwert diese Aufgabe
  - Wo sind die Agenten zurzeit?
  - Ist der Agent noch dort, wo er lokalisiert wurde?



Pfadkonzept



Proxy (Stellvertreter)
leitet Nachrichten an den
Agenten weiter

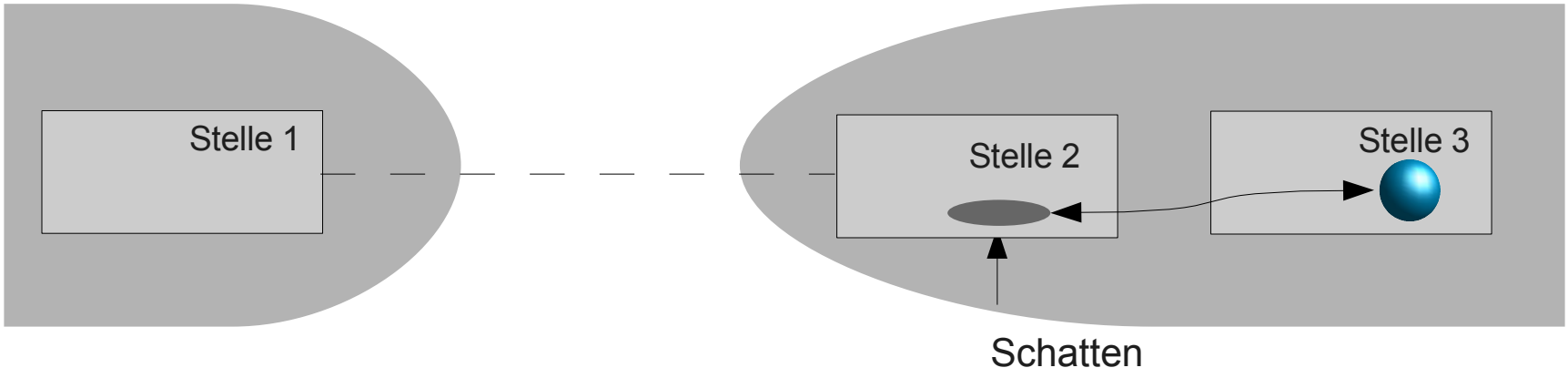
Aufwand zum Finden des Agenten: O(Pfadlänge)

Verfügbarkeit nimmt mit Länge exponentiell ab
(steigende Anzahl von Fehlerquellen)

Zu beachten:
Agent darf nicht
schneller als die
Nachricht migrieren

Schattenkonzept

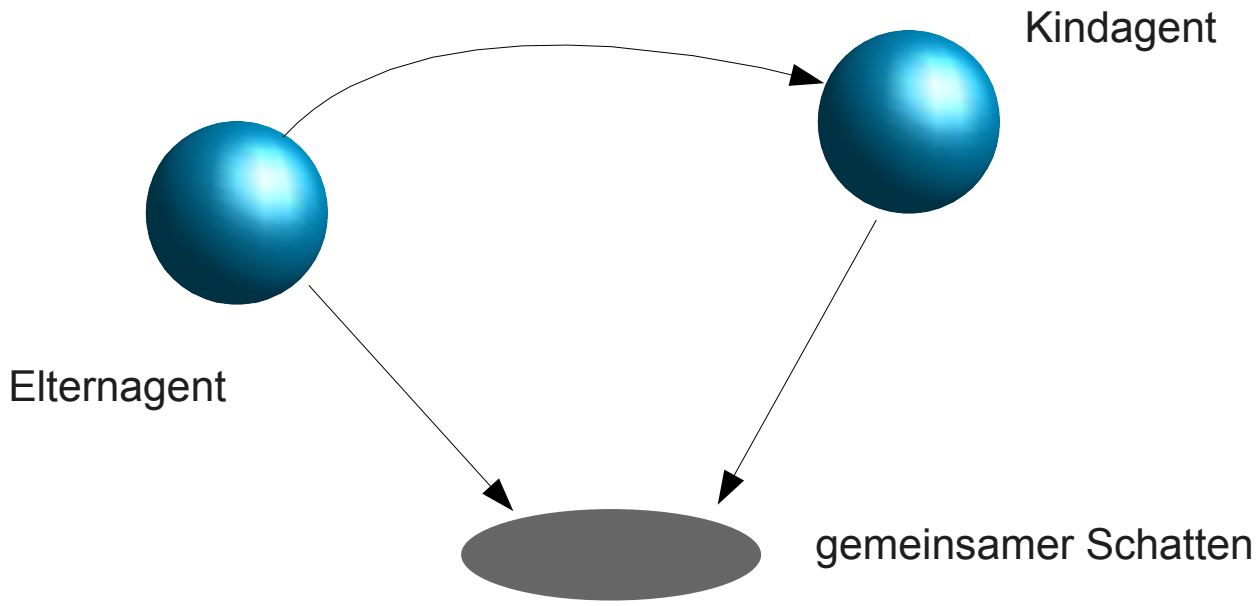
- Kombiniertes Energie- und Pfadkonzept
- Abhängigkeit zwischen Agent und seinem „Schatten“
  - Schatten ist Stellvertreter der Anwendung
  - kann auf anderem Knoten als die Anwendung installiert sein
  - vermindert Nichterreichbarkeit



Schattenkonzept

- In festgelegten Zeitabständen erfolgt Prüfung, ob der Schatten noch existiert
  - Waisenerkennung möglich
  - Anforderung eines neuen Zeitquantums durch den Agenten
- In der Überprüfungsphase darf der Agent nicht migrieren
  - Annahme: begrenzte Nachrichtenlaufzeit
- Schatten nicht erreichbar
  - Anwendung hat Schatten entfernt / Knoten abgestürzt
  - nach n Wiederholungen: Agent wird als Waise erkannt und terminiert

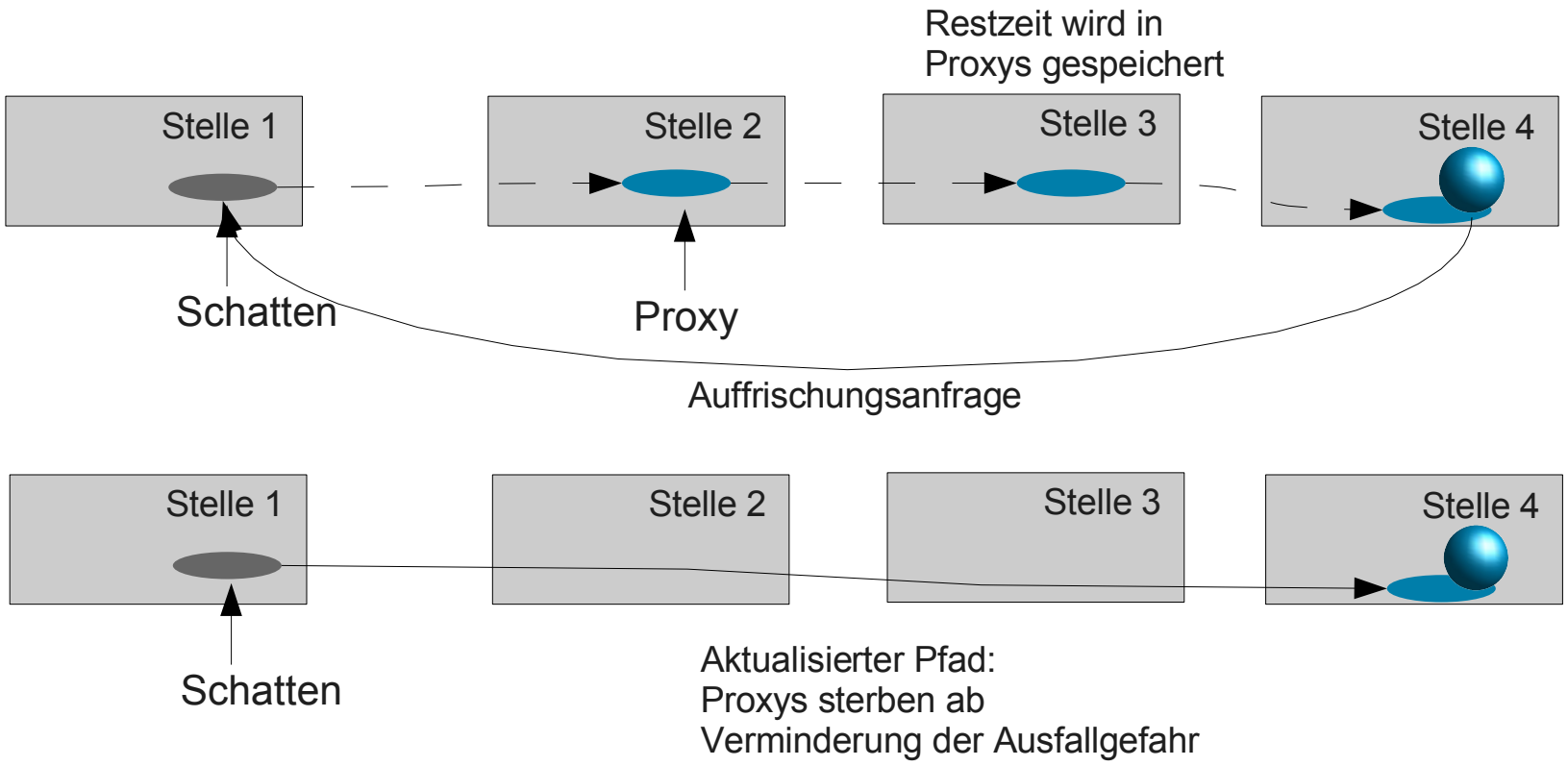
Erzeugen neuer Agenten



Verbleibende Restzeit bis zur Auffrischung wird kopiert
(damit also keine Chance auf Verlängerung durch Kopieren)

Schattenkonzept

- Hinzufügen des Pfadkonzepts, um Terminierung zu ermöglichen

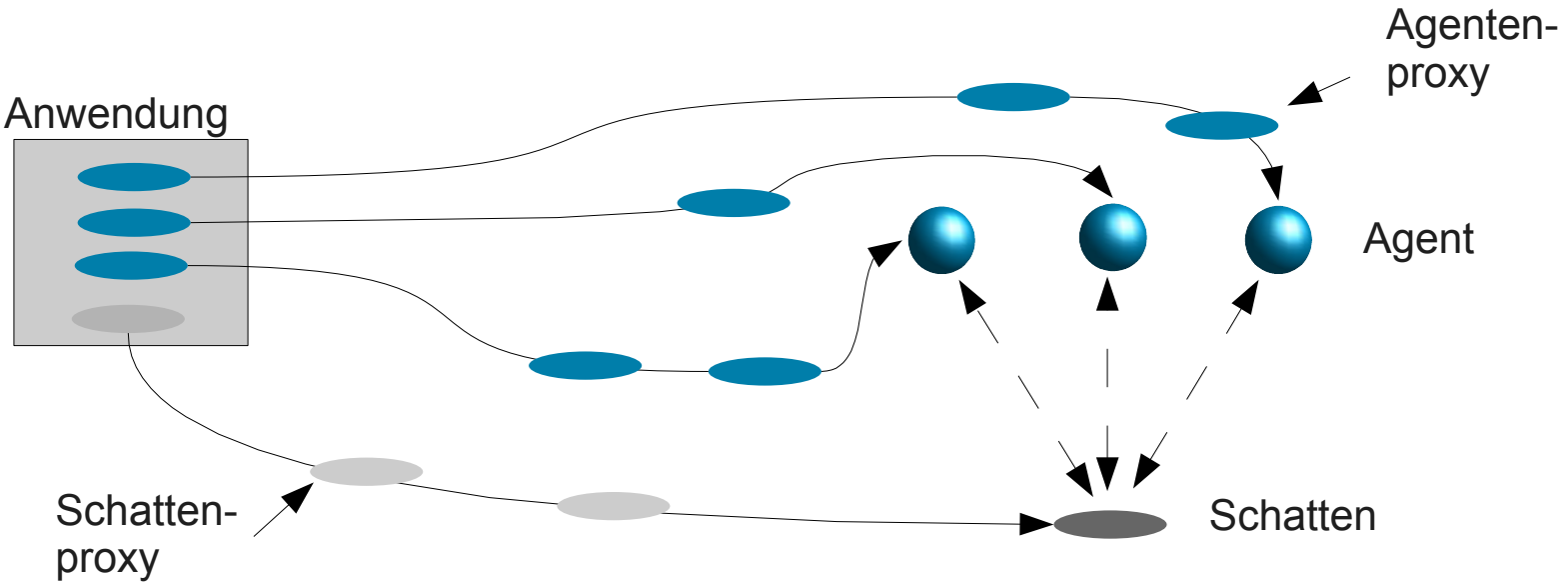


Bewegliche Schatten

- Gruppe von Agenten bewegt sich anwendungsabhängig
  - a priori nicht klar, wohin der Schatten platziert werden soll
  - Schatten muss sich auch bewegen können
- Mögliche Positionen
  - Minimierung der Kommunikationskosten
  - Auf der gleichen Stelle, an der ein relevanter Agent sitzt
    - Bei Absturz wird die gesamte Anwendung sicher beendet
- Schatten-Wanderung muss propagiert werden
  - an abhängige Agenten
  - an die Anwendung

Schattenproxys

- Analoges Vorgehen zu den Agentenproxys
  - Pfad von der Anwendung zum Schatten über dessen Proxys
  - Veraltete Proxys löschen
  - Ggf. auch maximale Pfadlänge festlegen



Schattenkonzept

- Verwandt mit Konzepten zur Entsorgung von Objekten
  - Garbage collection
- Anderes Fehlermodell bei Agenten
  - eigenständige Komponenten
  - können Ausführungsort wechseln
- Terminieren
  - direkt auf Befehl der Anwendung über Agentenproxy-Pfad
  - indirekt durch Verwaisung – über Energieverlust aufgrund Abtrennung vom Schatten

Andere Ansätze

- MAF
  - Management als Grundfunktion (inkl. IDL-Schnittstellen)
    - Starten, Stoppen, Lokalisieren von mobilen Agenten
  - Bedingungen an das Agentensystem
    - Buchführung
    - ggf. Einschränkung der Agentenautonomie (während bestimmter Vorgänge)
- Anwendungsorientiert
  - AMETAS
    - Management ist anwendungsspezifisch
    - Anwendungen müssen Management explizit vorsehen
    - kritisch gerade in früheren Phasen (viele Fehler)

Kritik

- „Hausgemachtes“ Problem
  - inhärentes Problem mobiler Agenten: einerseits mobil, andererseits autonom
  - Allgemeiner Ansatz
    - aufwändig
    - nicht unbedingt sicher
  - Auf Anwendungsebene besser lösbar?