FIPA

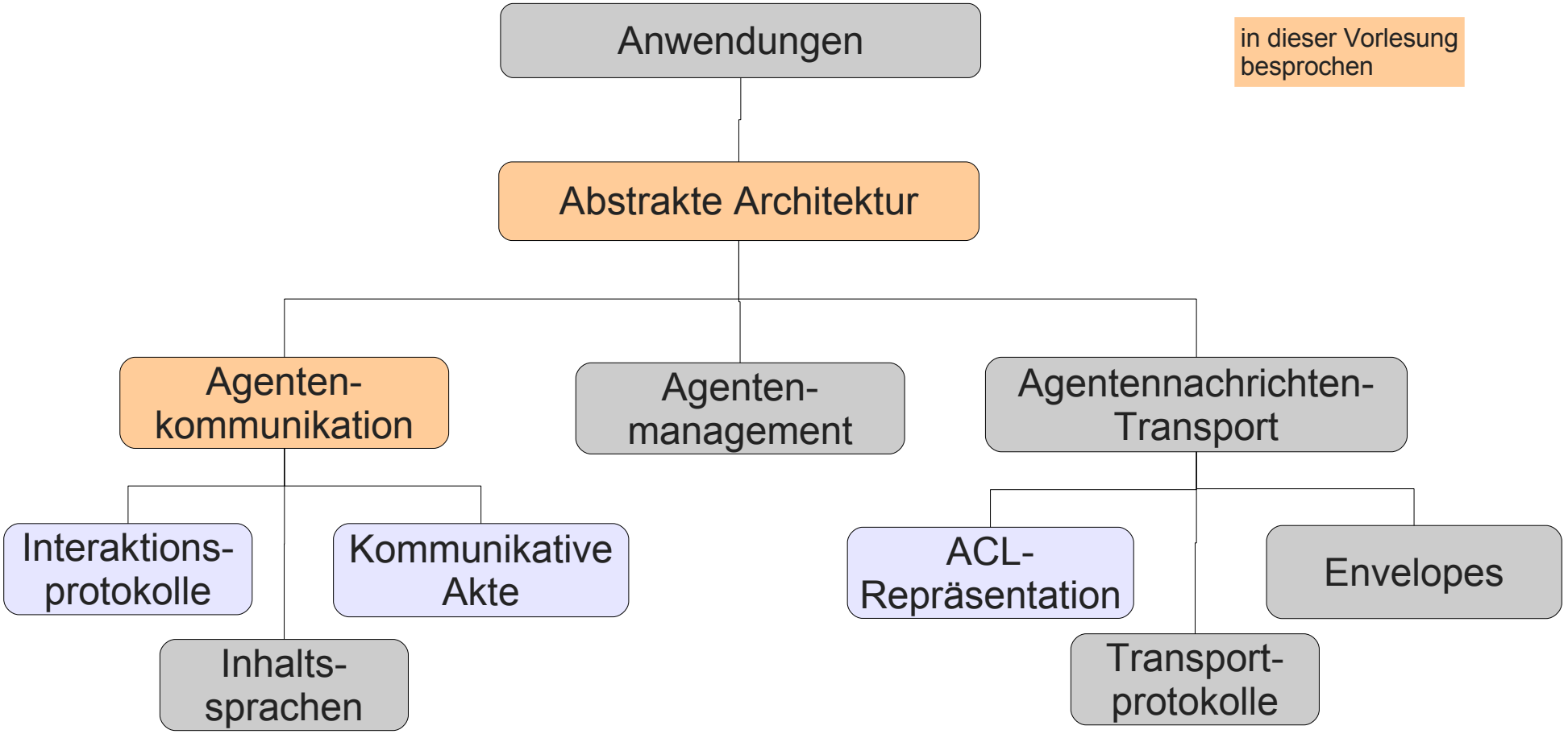
Foundation for Intelligent Physical Agents

Abstrakte Architektur
Agentenkommunikation

FIPA

- FIPA: Foundation for Intelligent Physical Agents: www.fipa.org
- Gegründet 1996
- Mission:
  - *The promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings.*
- Standardisierungsgremium
  - Agentenkommunikationsprachen
  - Inhaltssprachen
  - Architektur eines Agentensystems

Standardisierung nach FIPA



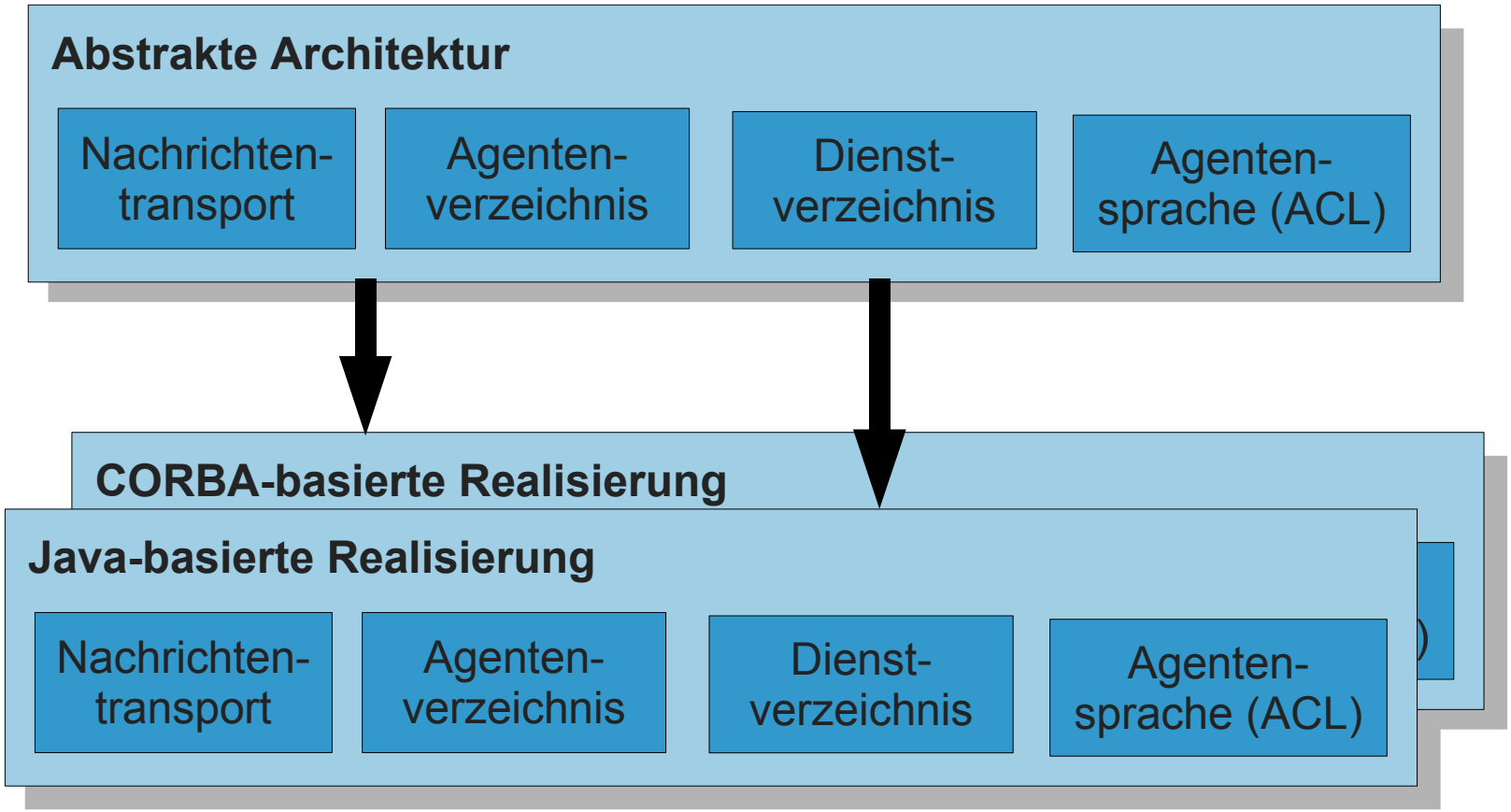
Spezifikationen

SC0001	FIPA Abstract Architecture Specification
PC0089	FIPA Policies and Domains Specification
SC0037	FIPA Communicative Act Library Specification
SC0061	FIPA Message Structure Specification
XC0086	FIPA Ontology Service Specification
SC0008	FIPA SL Content Language Specification
XC0009	FIPA CCL Content Language Specification

...

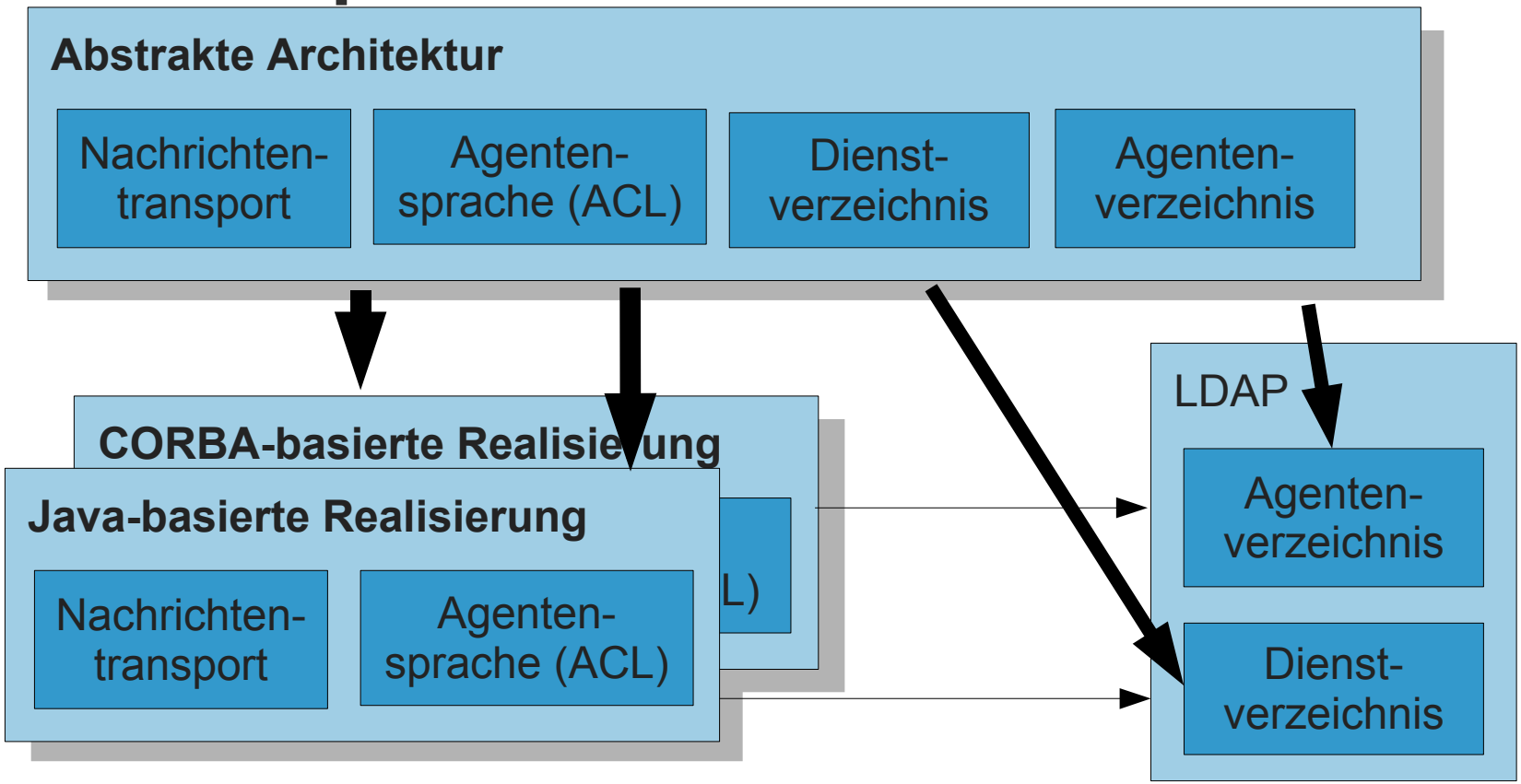
Abstrakte Architektur

Vom Abstrakten zum Konkreten



- FIPA schreibt nur die Mindestbedingungen vor
  - weit gehende Freiheiten für die Realisierung

Externe Komponenten



- Komponenten müssen nicht im abgeschlossenen System realisiert werden
- Verschiedene Systeme können sich Dienste teilen

Entstehung der FIPA-AAS

- Ziele der Abstrakte-Architektur-Spezifikation: Interoperabilität und Wiederverwendbarkeit fördern
- Notwendigkeit, gemeinsame Charakteristiken zu analysieren
  - daraus möglich, existierende Systeme auf das Vorhandensein dieser Charakteristiken zu überprüfen oder Parallelen zu ziehen
  - Beispiel: Abstraktionen des Nachrichtentransports
  - *Verpflichtende FIPA-Eigenschaften*
- Weitere Eigenschaften zu unterschiedlich
  - Sicherheitsmechanismen sind sehr von der Realisierung abhängig
  - *Optionale FIPA-Eigenschaften*

Ausrichtung der FIPA-Spezifikation

- Fokus auf Agenten-Interoperabilität
  - Verwaltung von mehreren Nachrichttransportschemata
  - Verwaltung von Nachrichtenkodierungsschemata
  - Auffinden von Agenten und Diensten über Verzeichnisdienste
- Keine agenteninternen Strukturen
- Keine Definition von
  - Agentensystem / -plattform
  - Gateways zwischen Systemen
  - Agentenkonfiguration und -koordinierung

Untrennbar mit der konkreten Realisierung verbunden

Spezifikation

- Festlegung einer abstrakten Architektur
  - Agenten können nicht als „Telescript“-ähnlich, als „AMETAS“-ähnlich, als „MAF“-kompatibel erklärt werden.
  - Stattdessen: Konzept des **Agenten** auf abstrakter Ebene
  - Spezifizierung einer Komponente anhand der Beziehung zu anderen Komponenten
- **Fett**/blau markiert
  - Abstrakte Vorrichtung, die eine bestimmte Funktionalität hat – wie auch immer realisiert.
  - Wenn z.B. ein AMETAS-Agent kein **Agent** ist, dann ist AMETAS nicht FIPA-kompatibel.
  - Zur Konformität mit der FIPA-AAS müssen alle verpflichtenden Eigenschaften von der konkreten Implementierung erfüllt werden.

FIPA-Namensraum

- Schlüssel-Wert-Paare (**key-value-pairs**)
  - Paare von Schlüsseln (**key**) und Werten (**value**).
  - Ein Paarelement (**pair-element**) besteht aus einer geordneten Liste von Token.
    - Traditionell: Textstrings
    - Natürliche (aber nicht ausschließliche) Repräsentation in der textuellen Form $Token_1$.$Token_2$.....$Token_n$ (punktierte Aneinanderreihung)
  - Schlüssel entstammen festem Namensraum (von FIPA verwaltet)
    - Ein Schlüssel ist stets ein Paarelement (pair-element);
    - Werte können Paarelemente sein; hängt vom Schlüssel und dem erwarteten Werteformat ab
    - Eigene Erweiterungen mit Präfix "x-"

FIPA-Namensraum

- Alle abstrakten Elemente der Spezifikation besitzen einen Namen in Form eines Paarelements
- Bei qualifizierten Paarelementen gibt es mehr als ein Token.
  - Das erste Token muss einer Toplevel-Domäne der ICANN entsprechen.
    - also de, uk, fr, ... , com, org, edu, ...
- Bei unqualifizieren Paarelementen wird ein vordefinierter Präfix angewendet
  - FQN(agent-name) = "org.fipa.standard.agent-name" (Fully qualified name)
  - FQN(agent-locator) = "org.fipa.standard.service.message-transport-service.agent-locator"
- Ein KVT (key-value-tuple) ist eine ungeordnete Menge von Schlüssel-Wert-Paaren

Übersicht aller abstrakten Elemente

Action-status
Agent
Agent-attribute (*)
Agent-communication-language
Agent-directory-entry
Agent-directory-service
Agent-locator
Agent-name
Content
Content-language
Encoding-representation
Encoding-service
Envelope
Explanation (*)
Message
Message-transport-service
Ontology (*)
Payload

Service
Service-address
Service-attributes (*)
Service-directory-entry
Service-directory-service
Service-name
Service-location-description
Service-locator
Service-root
Service-signature
Service-type
Signature-type
Transport
Transport-description
Transport-message
Transport-specific-address
Transport-specific-property (*)
Transport-type

(*) markieren optionale Elemente

Was sind Agenten, was sind Dienste?

- Aus der Spezifikation

> **Agents** communicate by exchanging messages which represent speech acts, and which are encoded in an **agent-communication-language.**

> **Services** provide support services for **agents**. [...] The Abstract Architecture is explicitly neutral about how **services** are presented. They may be implemented either as **agents** or as software that is accessed via method invocation, using programming interfaces such as those provided in Java, C++, or IDL.
> An **agent** providing a **service** is more constrained in its behaviour than a general-purpose agent. In particular, these agents are required to preserve the semantics of the service. This implies that these agents do not have the degree of autonomy normally attributed to agents. They may not arbitrarily refuse to provide the service.
> It should be noted that if services are implemented as agents there are potential problems that may arise with discovering and communicating with these services. The resolution of these issues is beyond the scope of this document.

Agenten und Dienste

- Agenten
  - kommunizieren mit Hilfe einer Agentenkommunikationssprache (ACL): **agent-communication-language**
  - haben einen besonderen Grad an Autonomie
  - können Verrichtung einer Dienstleistung ablehnen
- Dienste
  - bieten den Agenten Dienstleistungen an
  - Es wird erwartet, dass Dienste ihre Leistung auf Anfrage erbringen.
  - Agenten, die als Dienste auftreten, sind nicht mehr autonom im eigentlichen Sinne
  - Standarddienste: **agent-directory-service**, **message-transport-service**, **service-directory-service** müssen vorhanden sein

Agent

- Ein **Agent** ist ein Prozess, welcher die autonome, kommunikative Funktionalität einer Applikation implementiert
- Ein **Agent** kommuniziert in einer **Agentenkommunikations-sprache** (**agent-communication-language**, ACL)
- Jede FIPA-konforme Architektur **muss** das Konzept eines **Agent**en realisieren.
- Eigenschaften
  - Ein **Agent** besitzt einen Agentennamen (**agent-name**)
  - Ein **Agent** kann Attribute aufweisen (**agent-attributes**)
  - Ein **Agent** hat eine Lokalisationsbeschreibung (**agent-locator**), welcher die Transportbeschreibungen (**transport-description**) aufzählt

Agent

- Einem **Agenten** kann man eine Nachricht zusenden unter Verwendung eines Transportweges (**transport**), welcher in einer Transportbeschreibung (**transport-description**) genannt ist.
- Ein **Agent** kann einem anderen **Agenten** eine Transportnachricht zusenden (**transport-message**)
- Ein **Agent** kann sich bei einem oder mehreren Verzeichnissen anmelden (**agent-directory-services**), indem er einen Eintrag (**agent-directory-entry**) vornimmt.
  - Ein **Agent** darf seinen Eintrag modifizieren.
  - Ein **Agent** darf seinen Eintrag zurückziehen.
  - Ein **Agent** darf einen Eintrag in einem Verzeichnis suchen.
  - Ein **Agent** ist vermöge der Transportbeschreibungen in einem Verzeichnis adressierbar.

Agent

- Realisierung
  - Agenten können unterschiedlich realisiert sein
    - Java-Klasse(n)
    - COM-Objekt
    - Lisp-Programm
    - TCL-Skript
  - Ausführung kann unmittelbar oder interpretiert sein
- Verhältnis zwischen Agent und Umgebung
  - definiert durch den Lebenszyklus eines Agenten
  - aber nicht spezifiziert in FIPA, da zu inhomogen

Agentenname

- Agentenname, echter Name
  - **agent-name**: Schlüssel-Wert-Paar
  - opak (undurchsichtig), also ohne vorgegebene innere Struktur
  - Identifiziert den **Agent**en global eindeutig; Erzeugung muss die Eindeutigkeit möglichst sicher gewährleisten (ggf. Verwendung sicherer Zufallszahlen)
  - wird dem **Agent**en zur Erzeugungszeit zugeordnet und muss mit diesem verbunden bleiben
  - Existenz und Gültigkeit sollte nicht von erzeugender Instanz abhängen
  - Echtheit sollte nachweisbar sein

Agentenname

- Eigenschaften und Bedeutung
  - sollte keine semantischen Inhalte überbringen
  - sollte keine Hinweise auf Transportwege geben
  - sollte nicht zur Authentifikation verwendet werden (dazu sind weitere Informationen erforderlich, Zertifikate usw.)
  - kann das BDI-Modell unterstützen (später Genaueres hierzu)
    - z.B. kann der Agentenname Agenten kennzeichnen, denen bestimmte „Haltungen“ zu Aussagen (propositional attitudes) unterstellt werden
- Spitznamen (Nicknames)
  - kann zur einfachen Adressierung herangezogen werden, z.B. „ticket broker“
  - realisierbar als Attribut des Agenten (agent-attribute)
  - ersetzt nicht den eindeutigen Namen

Dienst

- Dienst (**service**)
  - Zusammenhängender Satz von Funktionalitäten zur Unterstützung von Agenten und anderen Diensten
  - Nicht zu verwechseln mit dem *Dienst*, der von einem Agenten als Teil seiner Aufgabe erbracht wird (= Dienstleistung)
- Komponenten und Eigenschaften
  - verfügt über öffentlich erkennbares und zugängliches Verhalten
  - verfügt über eine Beschreibung
  - kann von **Agent**en verwendet werden
  - beinhaltet **service-type**, **service-name**, **service-locator**
  - **Agent-directory-service, Message-transport-service, Service-directory-service** sind Pflichtinstanzen von Service
  - kann einen **Service-directory-entry** im **Service-directory-service** aufweisen (mit **service-type**, **service-name**, **service-locator**)

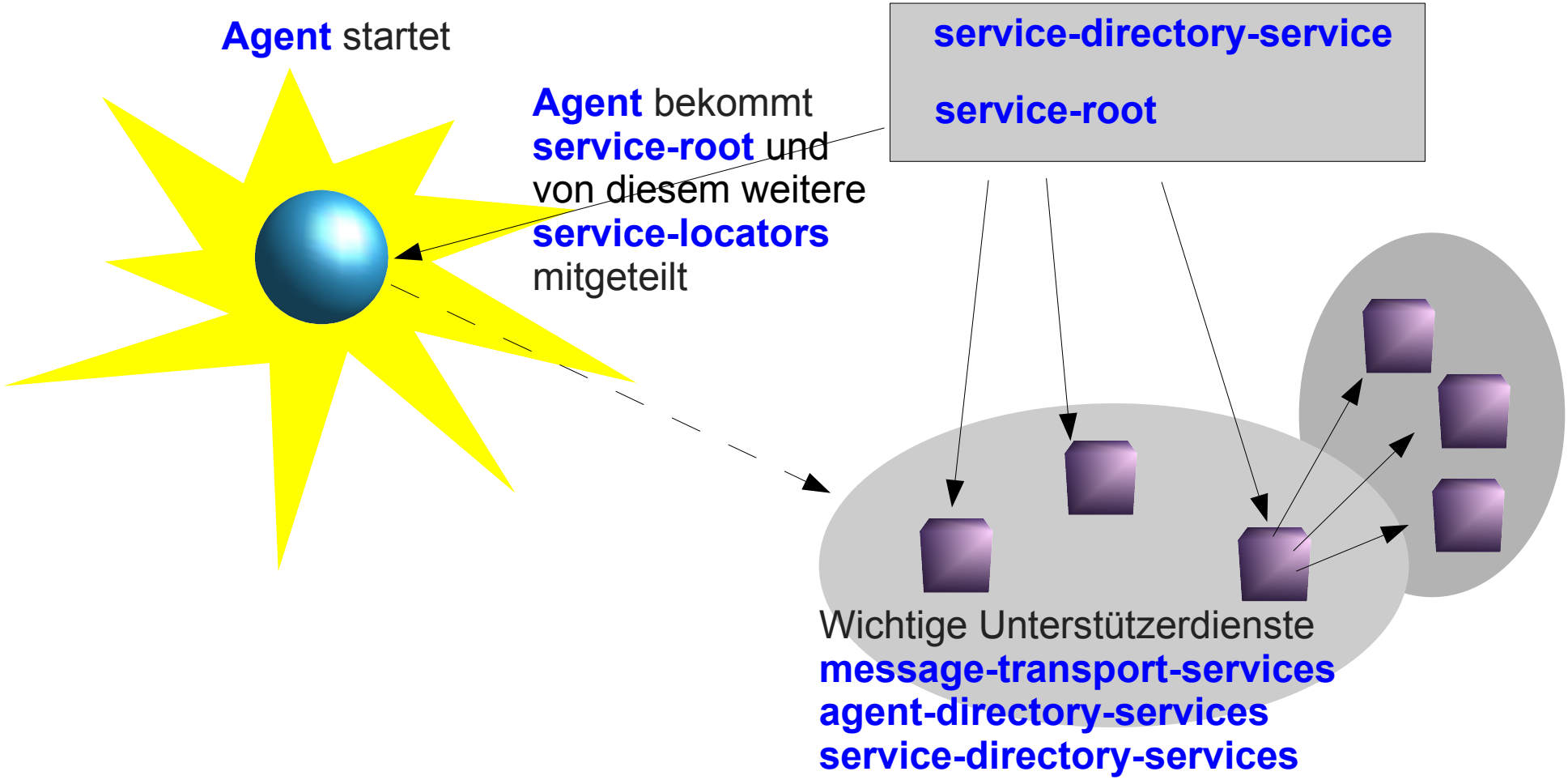
Verzeichnisdienste

- Vorrichtung **service-directory-service**
  - bietet **Agent**en und Diensten (**Services**) die Möglichkeit, Dienste (**Services**) zu finden
  - Bietet Diensten (**Services**) die Möglichkeit, sich zu registrieren.
  - Einträge **service-directory-entries**

service-name
service-type
service-locator

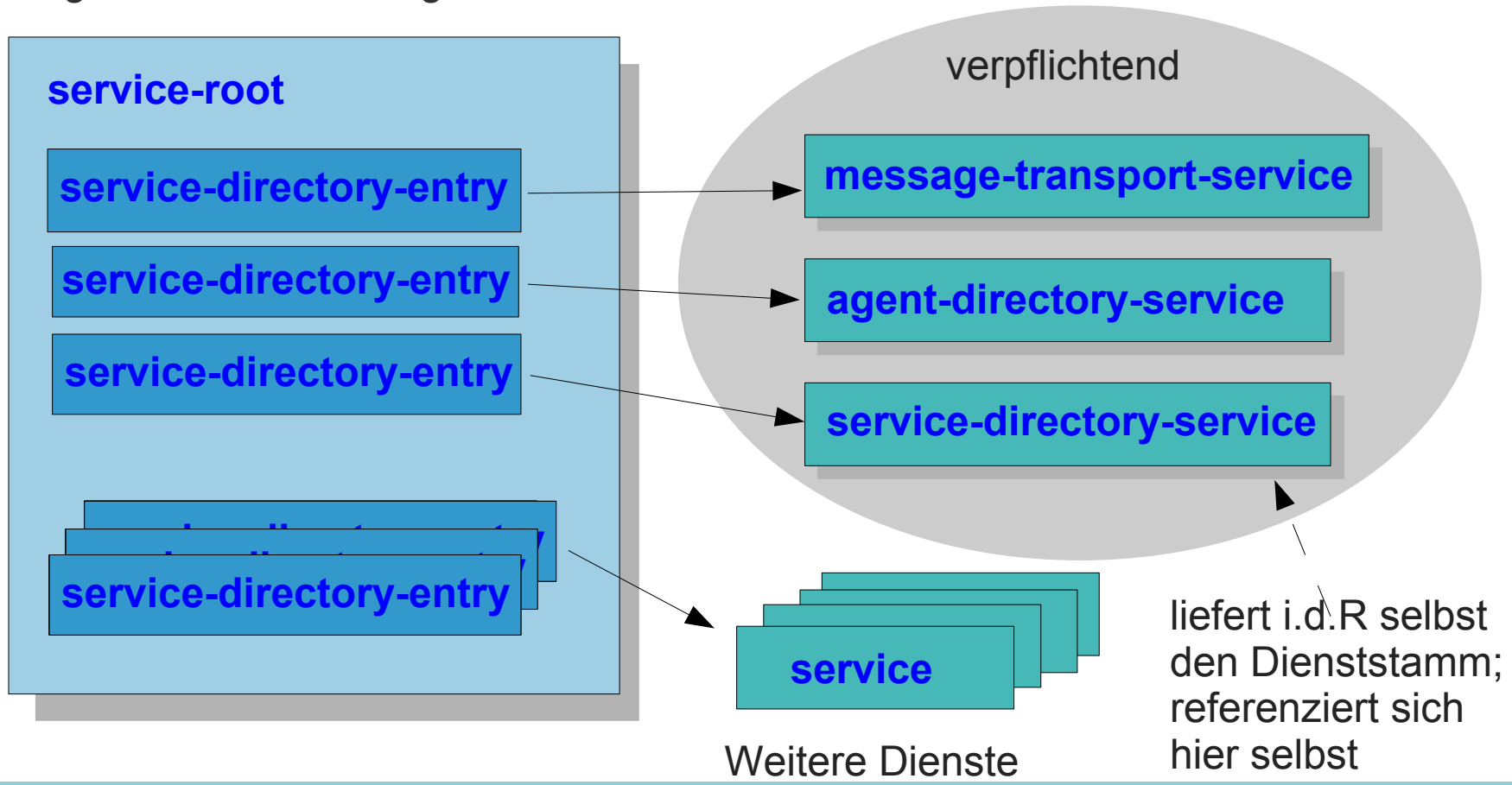
- Entsprechende Funktion: **agent-directory-service**
  - jedoch werden höhere Anforderungen an den Agentenverzeichnisdienst gestellt: ggf. per LDAP realisieren oder anderweitig verteilen
  - **service-directory-service** kann sogar auf einer festen Liste basieren

Starten eines Agenten



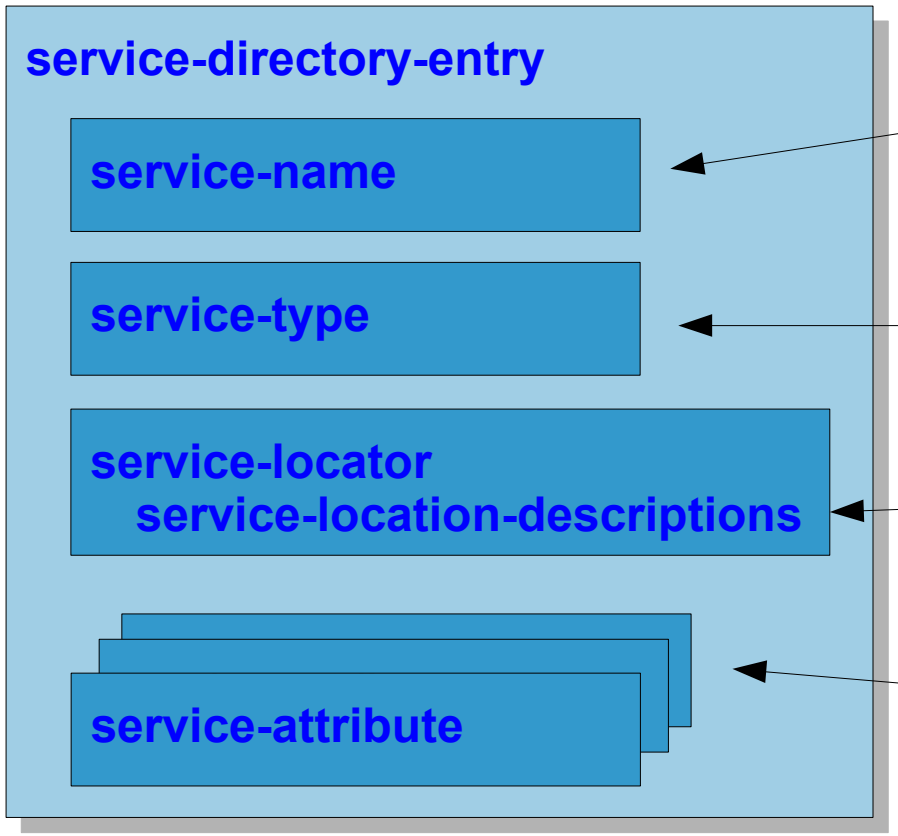
Dienststamm

- Dienststamm (**service-root**)
  - Menge von Diensteinträgen



Dienstverzeichniseintrag

- Dienstverzeichniseintrag (**service-directory-entry**)
  - verwaltet durch einen **service-directory-service**



Global eindeutiger Bezeichner

Schlüssel-Wert-Paare
in Bezug auf einen bestimmten
Namensraum

Adresse, Signatur usw.;
notwendig für einen **Agenten**, um
den **Service** anzusprechen

referenzieren z.B.
verschiedene
Instanzen

Dienste

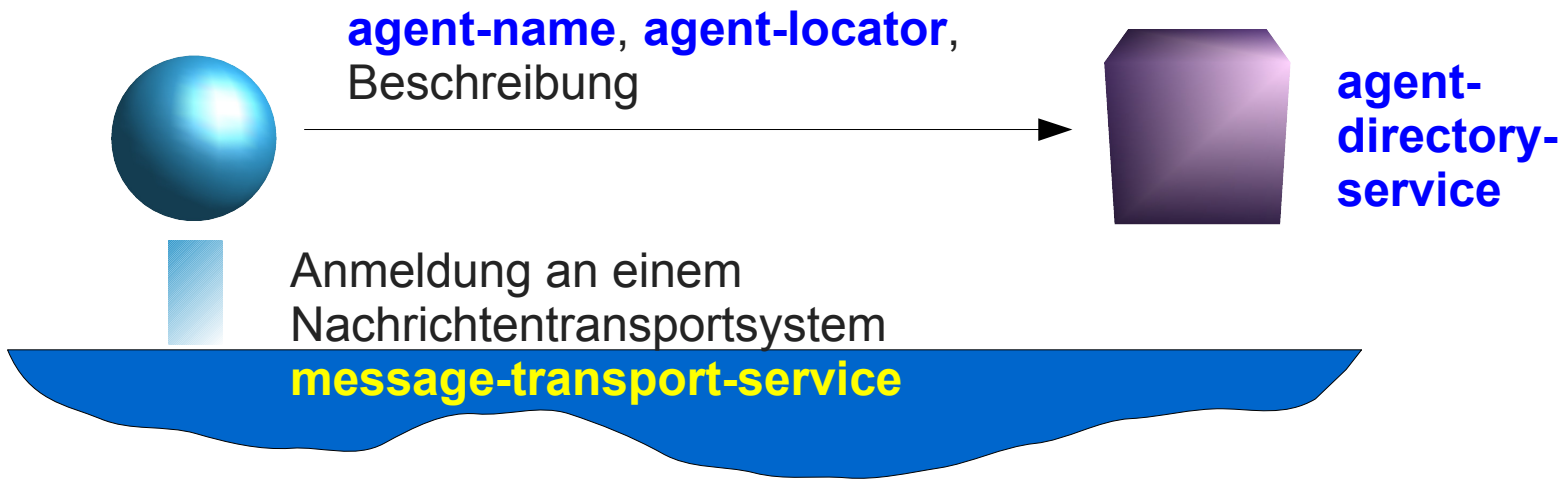
- Dienstadresse (**service-address**)
  - ein für den Diensttyp (**service-type**) spezifischer String
  - beschreibt die Adresse, unter der eine Bindung an den Dienst erfolgen kann (durch Agent oder anderem Dienst)
  - wird durch einen **signature-type** qualifiziert (z.B. "org.fipa.standard.service.java-rmi-binding")
- **service-locator**
  - Liste von Dienstadressen
  - Teil eines Service-directory-entry
  - wird von Agenten verwendet, um den Service anzusprechen

Agentenverzeichnis

- **agent-directory-service**
  - ermöglich einem **Agenten**, eine geeignete Beschreibung dort abzulegen (als **agent-directory-entry**).
  - ermöglicht anderen **Agenten**, nach einer bestimmten Beschreibung zu fragen, um mit diesen **Agenten** zu interagieren.
- **agent-directory-entry**
  - besteht aus **key-value-pairs**
  - **agent-name**: Global eindeutiger Bezeichner für diesen **Agenten**
  - **agent-locator** beinhaltet Liste von **transport-description**
  - ggf. weitere Infos
    - Dienstleistung
    - Kosten
    - Restriktionen

transport-type
transport-specific-address
[transport-specific-properties]

Registrieren eines Agenten

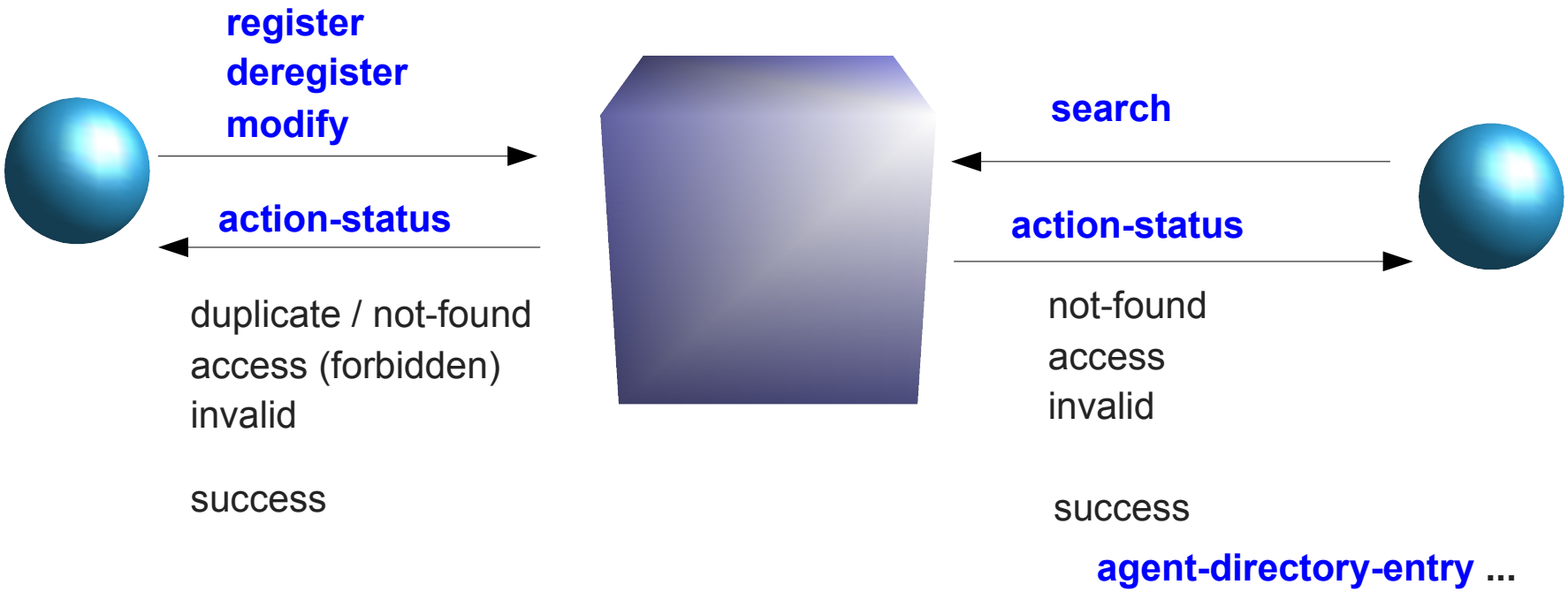


→ Definition des **transport**
Damit Adressierbarkeit

- Nach Registrierung ist der Agent von anderen Agenten aus erreichbar
  - das heißt: Man kann mit ihm kommunizieren

Verzeichnisdienste

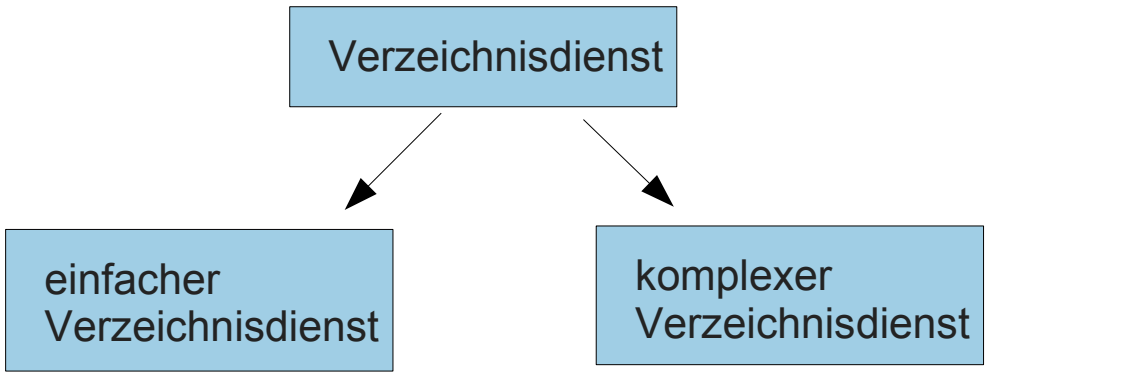
- Funktionen



FIPA spezifiziert nicht die Art der Lieferung mehrfacher Ergebnisse oder deren Auswahl

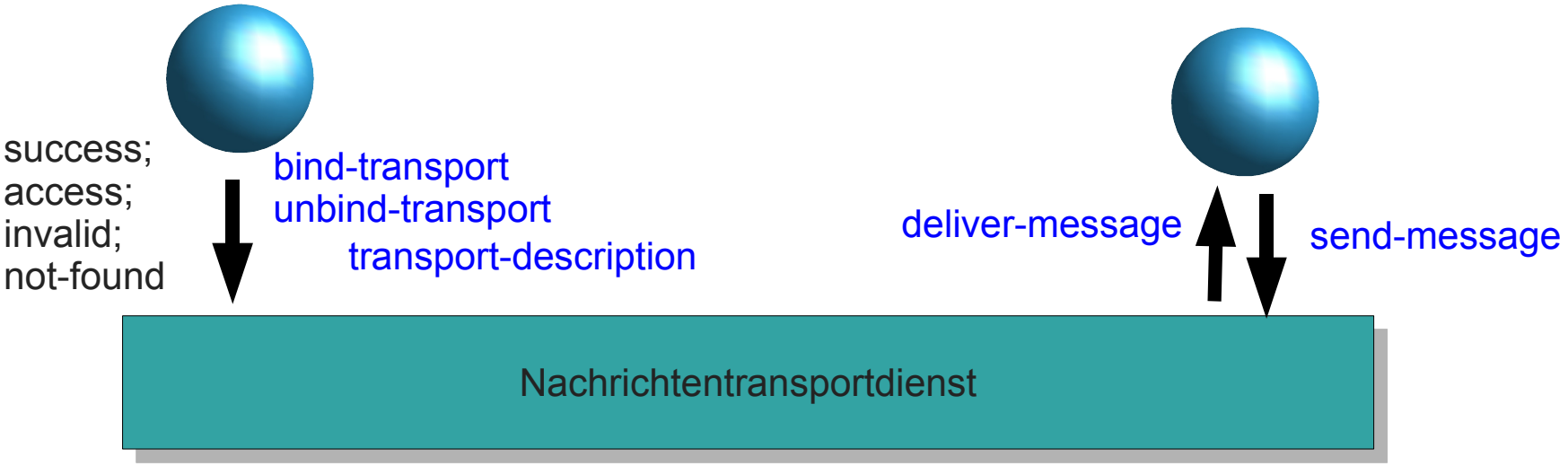
Erweiterte Verzeichnisdienste

- Viele existierende FIPA-Realisierungen besitzen alternativen Verzeichnisdienst **directory-facilitator**
  - verlangt jedoch zu mächtige Suchfunktionen für gängige Verzeichnisdienste wie LDAP, X.500, NIS
- Zweiteilung der Verzeichnisdienste



Nachrichtentransportdienst

- Nachrichtentransportdienst (**message-transport-service**)
  - Dienst (**service**), welcher Transportnachrichten entgegennimmt (**transport-message**), die zwischen **Agent**en verschickt werden
  - verpflichtende Komponente; jedoch müssen nur grundlegende Eigenschaften realisiert sein
    - Anbinden, lösen, verschicken, empfangen
    - nicht notwendig, verschiedene Transportwege zu realisieren

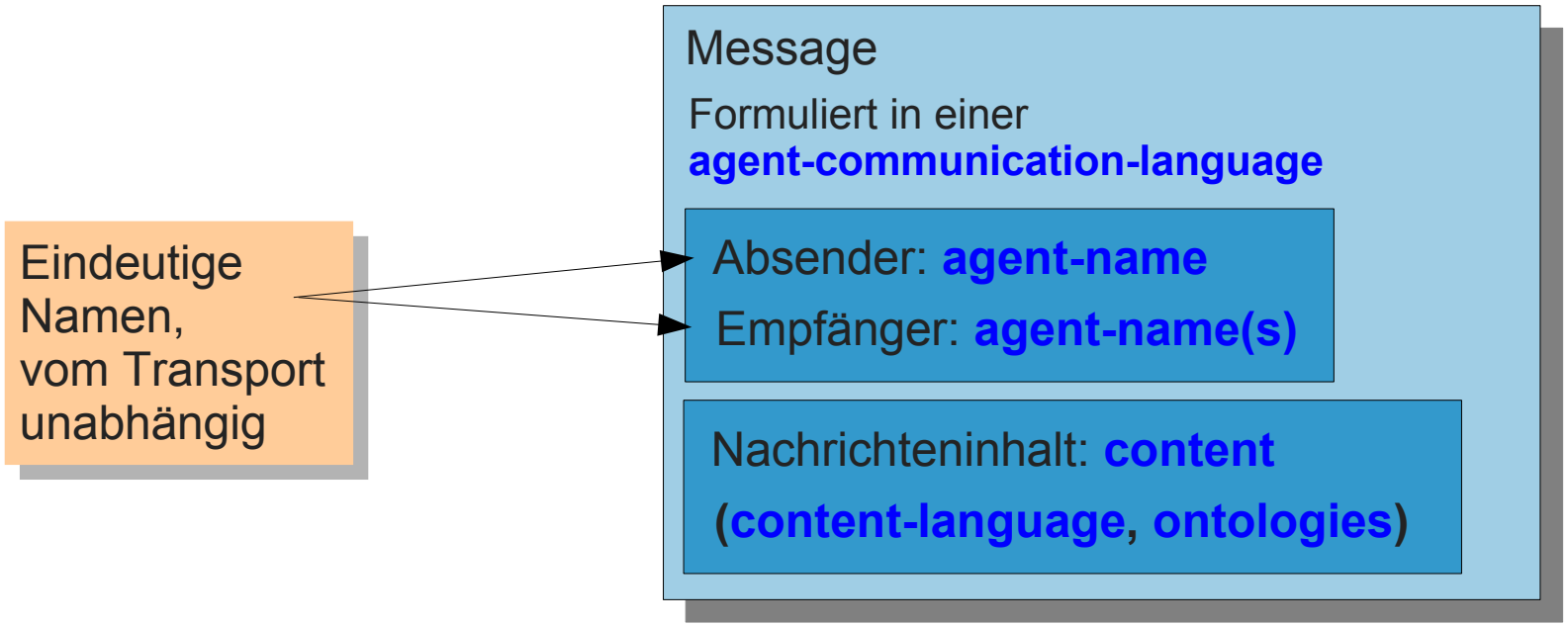


Nachricht

- Nachricht (**message**)
  - Struktur: KVT (also Menge von Schlüssel-Wert-Paaren)
  - Kommunikationseinheit zwischen zwei oder mehreren **Agent**en
  - manifestiert den kommunikativen Akt
  - kann andere Nachrichten (**message**) beinhalten
  - formuliert in einer **agent-communication-language**
  - ist **payload** einer **transport-message**
  - wird durch einen **encoding-service** zu **payload** oder entsteht aus **payload**

Nachricht

- Komponenten
  - Absender/Empfänger – echte Namen (**agent-names**)
  - Hinweis auf den kommunikativen Akt (z.B. `request`, `inform`)
  - Bezeichnung der Ontologien (**ontology**)
  - Nachrichteninhalt (**content**)



Nachrichteninhalt

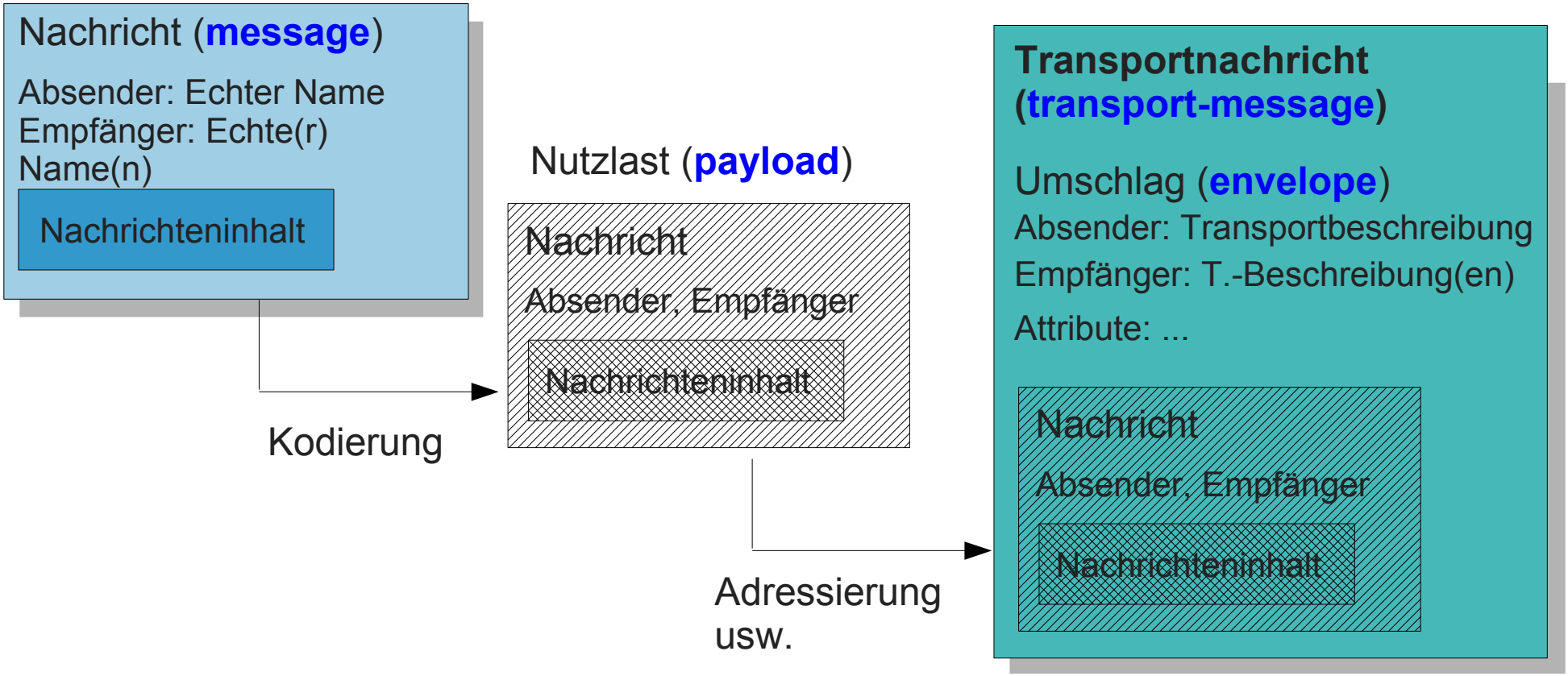
- Inhalt der Nachricht (**content**)
  - betrifft nicht den Transport
  - liefert Informationen über ein bestimmten Themengebiet
  - formuliert in einer Inhaltssprache (**content-language**)
  - ist parametrisierbar durch Ontologien (**ontology**)
  - ist verpflichtend zu realisieren: Es muss für die Erzeugung von Nachrichten möglich sein, einen Inhalt zu definieren.
- Beispiele für Inhaltssprachen
  - FIPA-SL
  - FIPA-RDF
  - FIPA-KIF
  - FIPA-CCL

ACL

- Agentenkommunikationssprache (**agent-communication-language**, ACL)
  - im Unterschied zur **Inhalts**sprache
  - erlaubt die Formulierung kommunikativer Akte
  - dient so zur Formulierung von Nachrichten (**messages**)
- Jede Realisierung der abstrakten Architektur muss eine ACL implementieren
- Genauere Betrachtung der *FIPA-ACL* im Anschluss

Keine spezifischen Angaben über die Natur der ACL

Nachrichtentransport

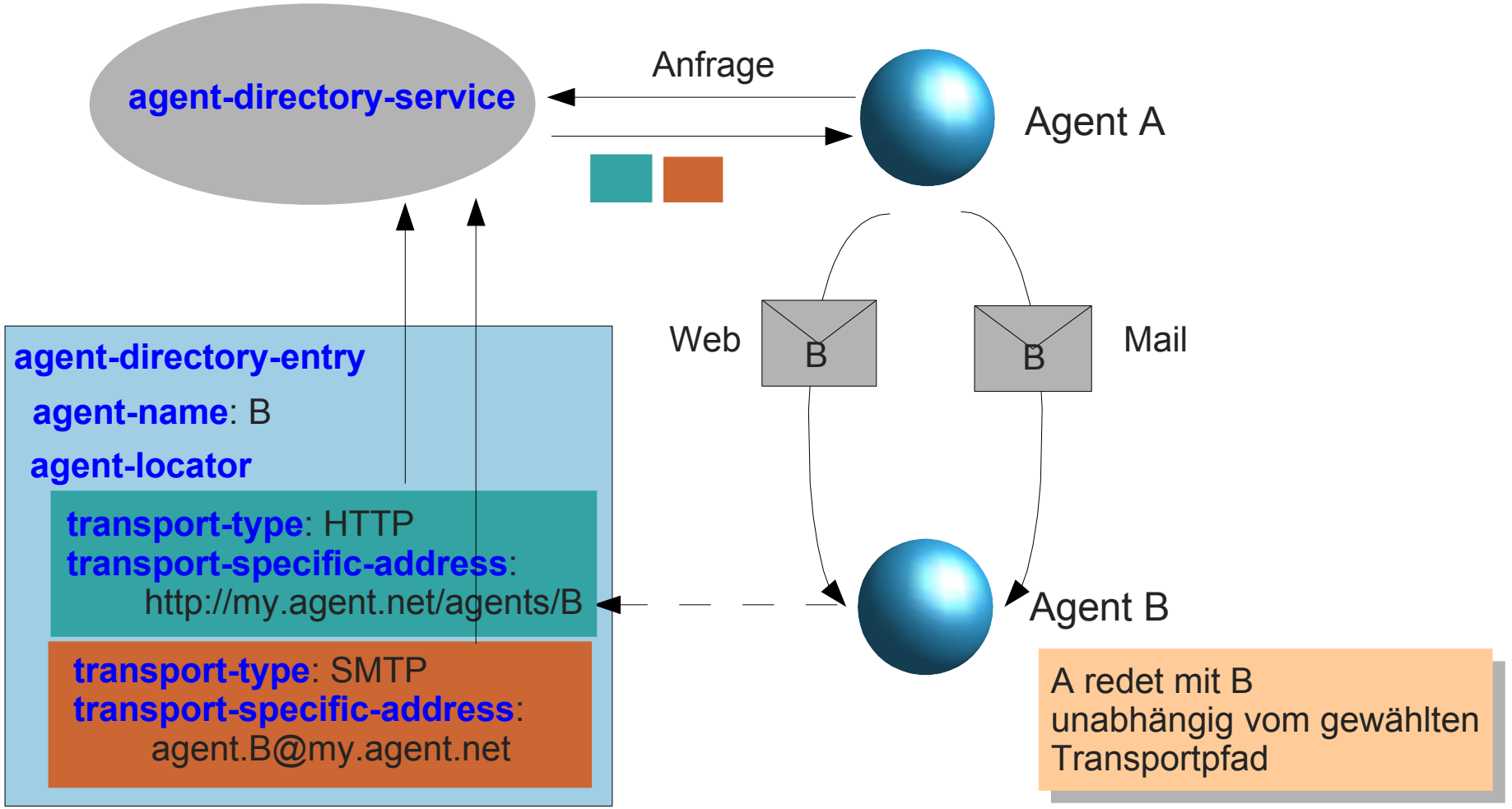


- Verwendung einer Transportbeschreibung im Gegensatz zum „echten Namen“ (**agent-name**)

Transportbeschreibungen

- Beschreibung der Parameter für den Transport der Nachricht
- Elemente
  - **transport-type**: HTTP, IIOP, SMTP, anderes
  - **transport-specific-address**: Webadresse, Socket/Repository-Bezeichner, Mailadresse, ...
  - **transport-specific-property**: Spezielle Parameter je nach Transportart
- Agentenname (echter Name, **agent-name**) muss nicht Teil der Transportbeschreibung sein!

Mehrere Transportbeschreibungen



Transportbeschreibungen

- Großer Vorteil der Transportbeschreibungen:
  - Adressierung erfolgt unabhängig vom Namen.
  - Logische Verarbeitung muss nicht den Transport beachten.
  - Agent braucht keine mehrfachen eindeutigen IDs; Agentenname muss alleine eindeutig sein.
  - Agent kann an anderer Stelle untergebracht werden.
  - Analogie mit Personenname und Adresse

Verschlüsselung

Transportnachricht: HTTP

Absender
Transport-Type: FIPA-HTTP
Transport-Address: http://...
Transport-properties: none

Empfänger
Transport-Type: FIPA-HTTP
Transport-Address: http://...
Transport-properties: Encrypt-3DES

Additional-attributes: none

Nachricht

Absender, Empfänger

Nachrichteninhalt

Verschlüsselung der Nutzlast

Transportnachricht: HTTP

Absender
Transport-Type: FIPA-HTTP
Transport-Address: http://...
Transport-properties: none

Empfänger
Transport-Type: FIPA-HTTP
Transport-Address: http://...
Transport-properties: Encrypt-3DES

Additional-attributes:
Public-key: ...
Payload-stat: 3DES-encrypt

Agentenkommunikation

FIPA-ACL

- FIPA-ACL wesentlich durch KQML beeinflusst
- Ähnliche Struktur, leicht abgewandelte Performative
- Jedoch gibt es Protokolle, die Ablauf definieren
  - Vordefinierte Szenarien: Auktionen, Verhandlungen usw.
- FIPA ACL schreiben Inhaltsstruktur vor
  - damit greift sie in die Gestaltungsmöglichkeit einer Inhaltssprache ein
- FIPA ACL und KQML lassen sich nicht unmittelbar ineinander überführen.

FIPA-ACL

- Wesentlicher Unterschied: Semantik
  - KQML nur informell spezifiziert, Beschreibung in Englisch
  - FIPA-ACL bringt zu jeder Performative ein semantisches Modell mit
    - Vorbedingung der Performative
    - Nachbedingung
  - damit exakt definiert, wie der Agent reagieren soll
  - Formulierung in FIPA-SL (Semantic Language)

Nachrichtenstruktur

- Parameter einer Nachricht

performative	Art des kommunikativen Aktes
sender	Absender der Nachricht
receiver	Empfänger der Nachricht
reply-to	Empfänger der Antwort
content	Inhalt
language	Inhaltssprache
ontology	Ontologie des Inhalts
encoding	Kodierung des Inhalts
protocol	Protokoll der Kommunikation
conversation-id	Referenz der Konversation
reply-with	Referenz der Antwortnachricht
in-reply-to	Referenz der Ursprungsnachricht
reply-by	Zeitpunkt, bis zu dem die Antwort vorliegen muss

  - Erweiterungen sind möglich: `x-Parameter`

Nachrichtenstruktur

- Die tatsächliche Repräsentation ist nicht Gegenstand dieser Spezifikation (FIPA ACL Message Structure Specification)
  - Spezifikation vorhanden für
    - „Bit-efficient“:
      0x01 /* accept-proposal */
      0x02 /* agree */
      0x03 /* cancel */
      ...
    - XML (<accept-proposal>...</accept-proposal>)
    - String
      Notation wie KQML: (Performative {:Parameter Wert}*)
      Erster Wert ohne Parameter ist der Wert von *performative*

Parameter der ACL

- `performative`: Standardnamen von FIPA sollten verwendet werden
- `sender`: Kann weggelassen werden bei anonymen Nachrichten
- `receiver`: Kann einzelner Agent oder eine Menge sein (Multicast)
- `reply-to`: Antworten nicht an den `sender`, sondern hierhin
- `reply-with`: Ausdruck, der die Antwortnachricht kennzeichnen soll
- `in-reply-to`: Ausdruck, der die Ursprungsnachricht referenziert
- `content`: Objekt der Aktion. Empfänger muss den Inhalt selbst auswerten.
- `language`: (optional) Formale Inhaltssprache. Vordefinierte Werte: FIPA-SL, FIPA-CCL, FIPA-KIF, FIPA-RDF
- `encoding`: (optional) Bestimmt die Datenrepräsentation der Nachricht
- `ontology`: (optional) Definiert die Bedeutung der Symbole in der Nachricht

Protokollkonzept

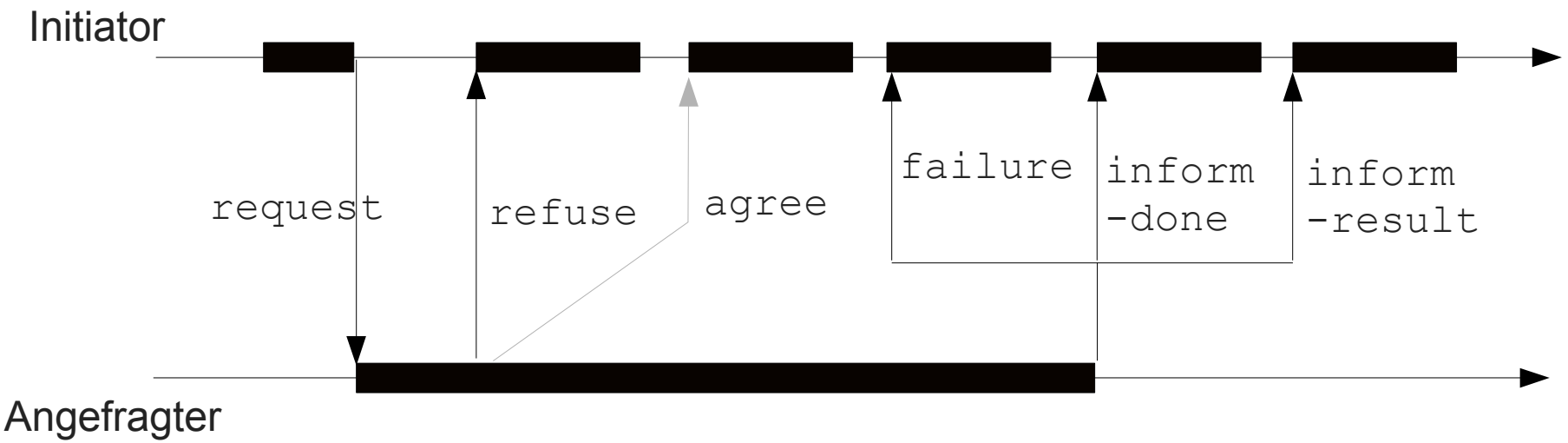
- `protocol`: (Optional) Interaktionsprotokoll.
  - Aber ohne `protocol` bestimmt der Inhalt das Protokoll (kann schwierig werden!)
  - Deutlicher Unterschied zu KQML, das keine Protokolle vordefiniert

Verwendung eines Protokolls macht die *Kommunikation* zur *Konversation*.

- `conversation-id`: Identifikator der Konversation
  - Muss von Initiator des Protokollablaufs gesetzt werden.
  - Muss stets wieder zitiert werden.
  - Muss global eindeutig sein
- `reply-by` bestimmt die Maximalzeit bis zum nächsten Protokollschritt (oder allgemein bis zur Antwort)

Protokollbeispiel

- Einfaches Beispiel: Request-Protokoll (`fipa-request`)



`agree`: Wenn Benachrichtigung erforderlich und Zustimmung vorliegt. Kann weggelassen werden, wenn die Abarbeitung schnell genug erfolgen wird (beachte gesetztes `reply-by`)

Protokolle

- Definierte Protokolle

fipa-request	Einfache Anfrage, ein Ergebnis zu liefern
fipa-query	Abfrage einer Wahr/Falsch-Bedingung oder nach einem Objekt
fipa-request-when	Wie request, aber auf eine Bedingung wartend
fipa-contract-net	Vorverhandlung eines request, dann request
fipa-iterated-contract-net	Wie contract-net, aber mit mehreren Runden
fipa-brokering	Vermittelte Anfrage (indirekte Anfrage/Antwort)
fipa-recruiting	Vermittelte Anfrage (indirekte Anfrage/direkte Antwort)
fipa-subscribe	Fortlaufende Aktualisierung der Antwort
fipa-propose	Vorschlag einer kommenden Aktion (durch den Initiator)

Weiterhin definiert: English auction, dutch auction (Anwendungsfälle)

Kommunikative Akte

Generelle Status	
agree	Ich führe die von dir gewünschte Operation aus.
refuse	Ich weigere mich, die von dir erwünschte Aktion durchzuführen.
failure	Ich konnte deine Anfrage nicht ausführen.
not-understood	Ich habe nicht verstanden, was du von mir wolltest.
Proposals	
cfp	Ich bitte um Angebote zu folgender Aktion.
propose	Ich biete an, die angefragte Aktion auszuführen.
accept-proposal	Ich nehme dein Angebot an.
reject-proposal	Ich lehne deinen Vorschlag ab.
Wissensaustausch	
inform	Glaube mir, dass Folgendes wahr ist.
inform-if *	Glaube mir, dass die Aussage wahr/falsch ist
inform-ref	Glaube mir: Der Wert von ... ist ...
confirm	Ich bestätige dir, dass Folgendes wahr ist.
disconfirm	Ich bestätige dir, dass Folgendes falsch ist.
propagate	Glaube mir Folgendes und leite diese Propagierung weiter.
subscribe	Schicke mir fortlaufend Aktualisierungen

Kommunikative Akte

Abfragen

query-if	Informiere mich, ob folgende Aussage wahr ist.
query-ref	Informiere mich über die Werte zu folgender Abfrage.
request	Führe folgende Aktion aus.
request-when	Führe folgende Aktion aus, wenn die Bedingung eintritt.
request-whenever	Wie request-when, aber wiederholt, wenn die Bed. wahr wird.
cancel	Kommando zurück, vergiss die request-Anfrage

Vermittlung

proxy	Leite die enthalt. Nachricht an die genannten Agenten weiter.

Sehr viel geringerer Sprachumfang als KQML

* Inform-If ist eigentlich nur eine Kurzschreibweise für Standard-Inform-Performative, die im Inhalt einer Anfrage auftauchen. Beispiel: Agent A bittet Agent B, ihn zu informieren, ob eine Aussage wahr oder falsch ist. Im „Request“ an B würde im Inhalt ein Inform-If auftauchen. B würde aber mit Inform (ob wahr oder falsch) antworten.
Auch Inform-Ref führt zu einer Kette von Informs.

Beispiele von FIPA-ACL-Nachrichten

```
(inform
    :sender (agent-identifier :name i)
    :receiver (set (agent-identifier :name j))
    :language Prolog
    :content "wetter(heute,regen)")
```

```
(failure
    :sender (agent-identifier :name j)
    :receiver (set (agent-identifier :name i))
    :language fipa-sl
    :content "((action (agent-identifier :name j)
                (open \"datei.txt\"))
                (error-message \"Datei nicht gefunden: datei.txt\"))")
```

Semantik von kommunikativen Akten

- Am Beispiel von inform und confirm

inform:

<i, inform(j,Φ)>

Vorbedingung: $B_i\Phi \wedge \neg B_i(Bif_j\Phi \vee Uif_j\Phi)$

Nachbedingung: $B_j\Phi$

$B_i\Phi$: i glaubt, dass Φ gilt
$U_i\Phi$: i hält Φ für möglich, weiß es aber nicht
$Bif_i\Phi = B_i\Phi \vee B_i\neg\Phi$

inform ist ein Akt, von i ausgehend, an j gerichtet, mit Aussage Φ.
Vorbedingung:
i glaubt, dass Φ gilt
und i glaubt nicht,
dass j entweder weiß, dass Φ gilt oder nicht gilt
oder dass j unsicher ist, ob Φ gilt oder nicht gilt
Nachbedingung:
j glaubt, dass Φ gilt.

Semantik von kommunikativen Akten

confirm:

<i, confirm(j,Φ)>

Vorbedingung: $B_i\Phi \wedge B_i U_j \Phi$

Nachbedingung: $B_j\Phi$

confirm ist ein Akt, von i ausgehend, an j gerichtet, mit Aussage Φ.
Vorbedingung:
i glaubt, dass Φ gilt
und i glaubt, dass j unsicher ist, ob Φ gilt.
Nachbedingung:
j glaubt, dass Φ gilt.

Kritik an FIPA

The FIPA proposal is not just for an agent communication language (ACL) but rather for an ACL **along with a theory of agency** (on top of which the ACL is defined). The two are inseparable both at the semantic and the implementation level. Changes at the "theory of agency" level would result to changes at the level of the semantic description of the ACL. **It is not feasible to expect that all agents that might use a *universal* ACL will comply with a single agent model.**

At the "theory of agency" level not everything is fine, as often suggested in the KQML-list threads, merely because the suggested framework is "formal". [...]

We would like to make it clear that we find the work behind the FIPA proposal for an ACL to be extremely interesting and important. The question is whether this is the way one wants to address the issues of the semantic description for an ACL **for *all* agents,** i.e., by irrevocably linking the "theory of agency" and the ACL layers. Furthermore, there is a myriad of issues regarding a *universal* ACL that are not addressed in the current FIPA proposal. The FIPA proposal only covers the most **rudimentary** message types and it remains to be seen how the suggested approach can address the other issues of interest to the agent community.

Finin und Labrou in http://www.cs.umbc.edu/kqml/papers/fipa/comments.shtml