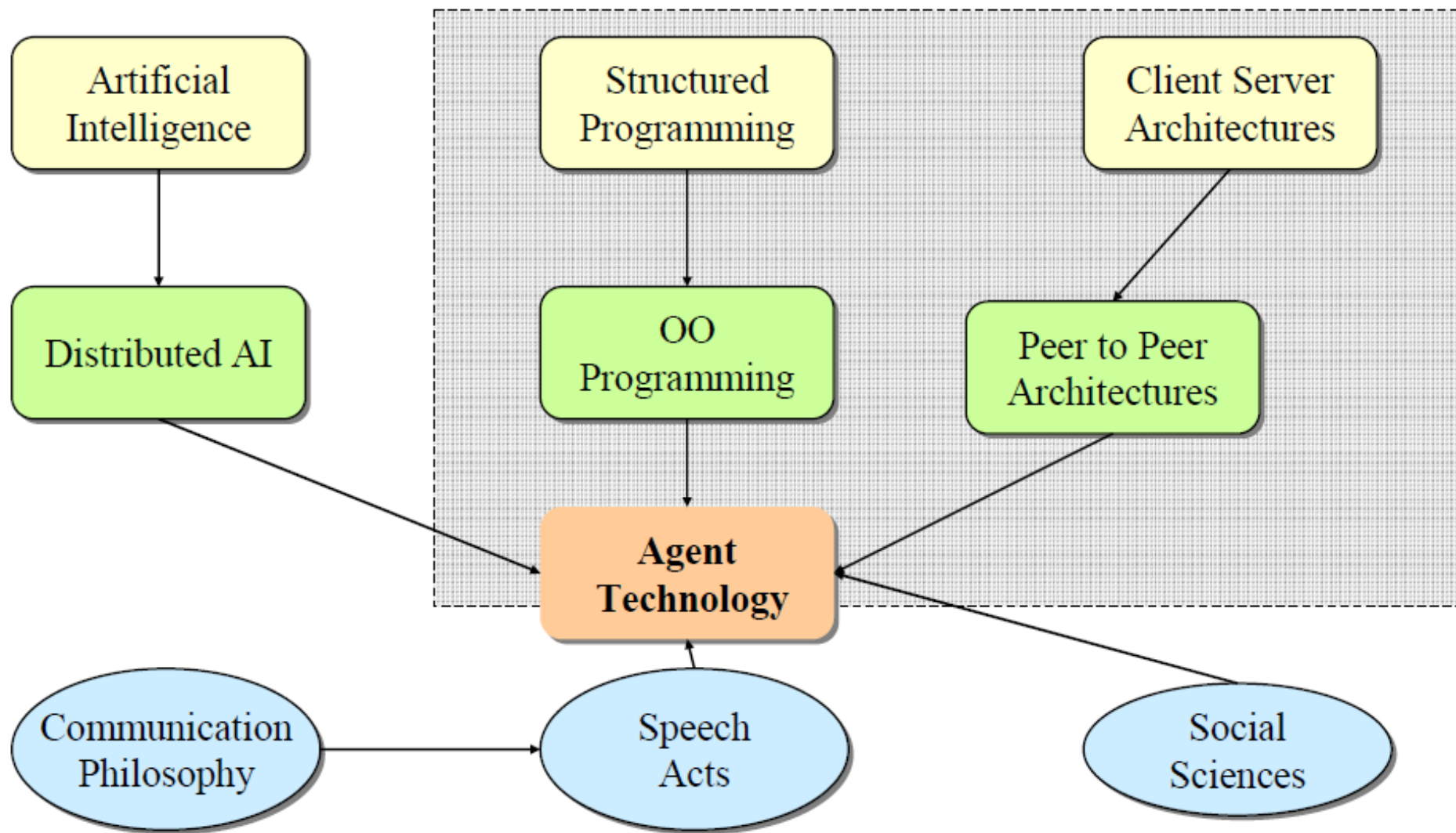


Спецификации IEEE FIPA. Архитектура на FIPA.

Аделина Алексиева-Петрова
aaleksieva@tu-sofia.bg

Агент-базирани технологии



Agent-Oriented Software Engineering (AOSE)

- Агент-базираният подход улеснява проектирането на комплексни системи, тъй като дава възможност за дефиниране на едно и също понятие (т. нар. агент) при анализ и проектиране на решение и неговото имплементиране.
- Разработката на софтуерна система с AOSE обикновено включва използването на агент-базирана мидълуер платформа и може да се реализира (но не задължително) с език за програмиране.

Agent-Oriented Software Engineering (AOSE)

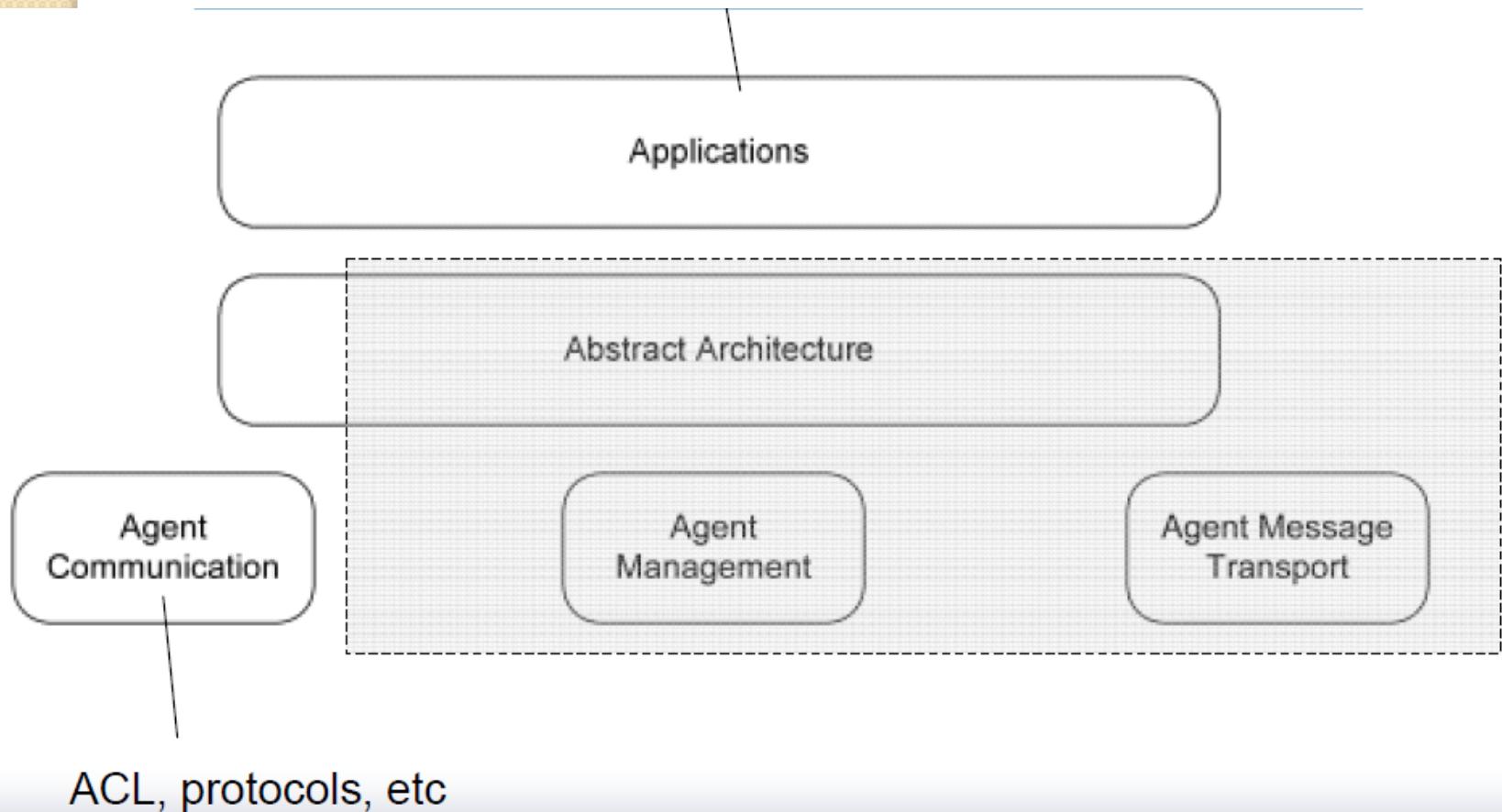
- От гледна точка на изпълнение, агенти са следващата стъпка в еволюцията на подходите за софтуерно инженерство и програмни езици.
- Това е стъпка за увеличаване степента на локализация и капсулиране в основните градивни елементи на програмните модели:
 - *Структури* (например в C)
 - *Обекти* (например в C++ и Java)
 - *Агенти*
- Агент е програмна единица, която е автономна в реално време и не е необходимо да бъдат пряко контролирани.

IEEE FIPA

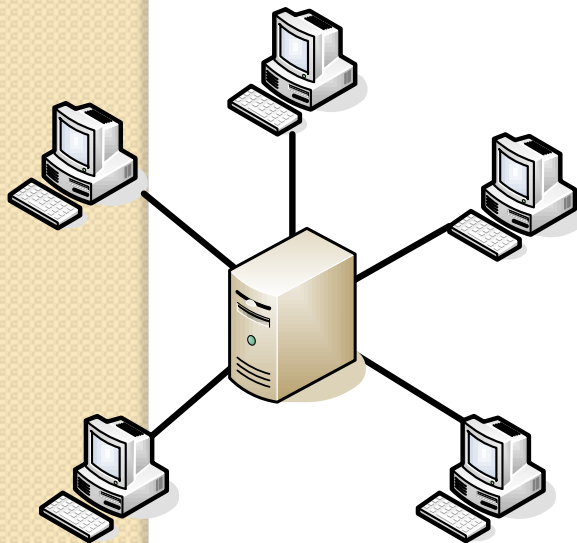
- Foundation for Intelligent Physical Agents (FIPA)
<http://www.fipa.org/>.
- FIPA е организация IEEE Computer Society, която насърчава агент-базирани технологии и оперативната съвместимост на нейните стандарти и други технологии.
- Основана през 1996.
- Разработва и стандартизира FIPA модел на MAS.

FIPA спецификации

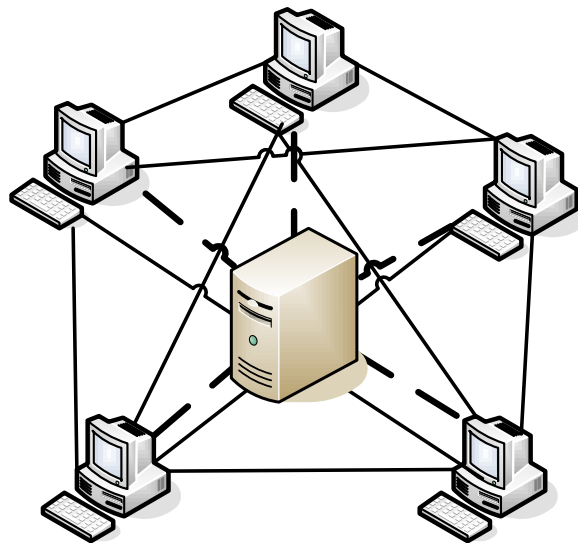
- Описание на агенти и онтологии за подпомагане на приложения



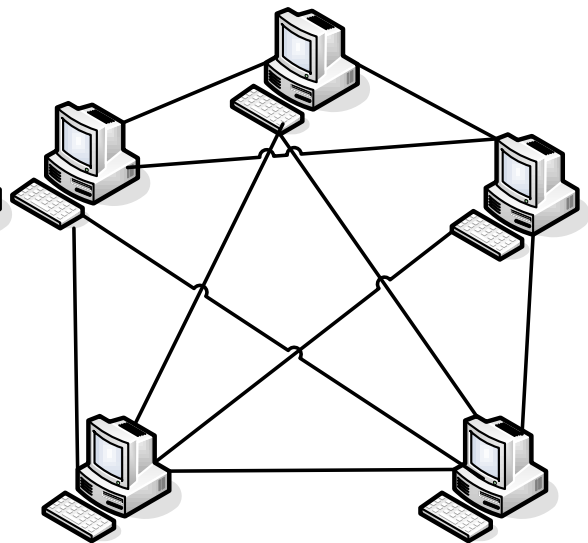
FIPA архитектура



Клиент/Сървър



Хибриден P2P



P2P

Организация на комуникацията на FIPA

Взаимодействащ
протокол

Комуникативно действие

Изразяване на
съдържание

Онтология

Съобщения

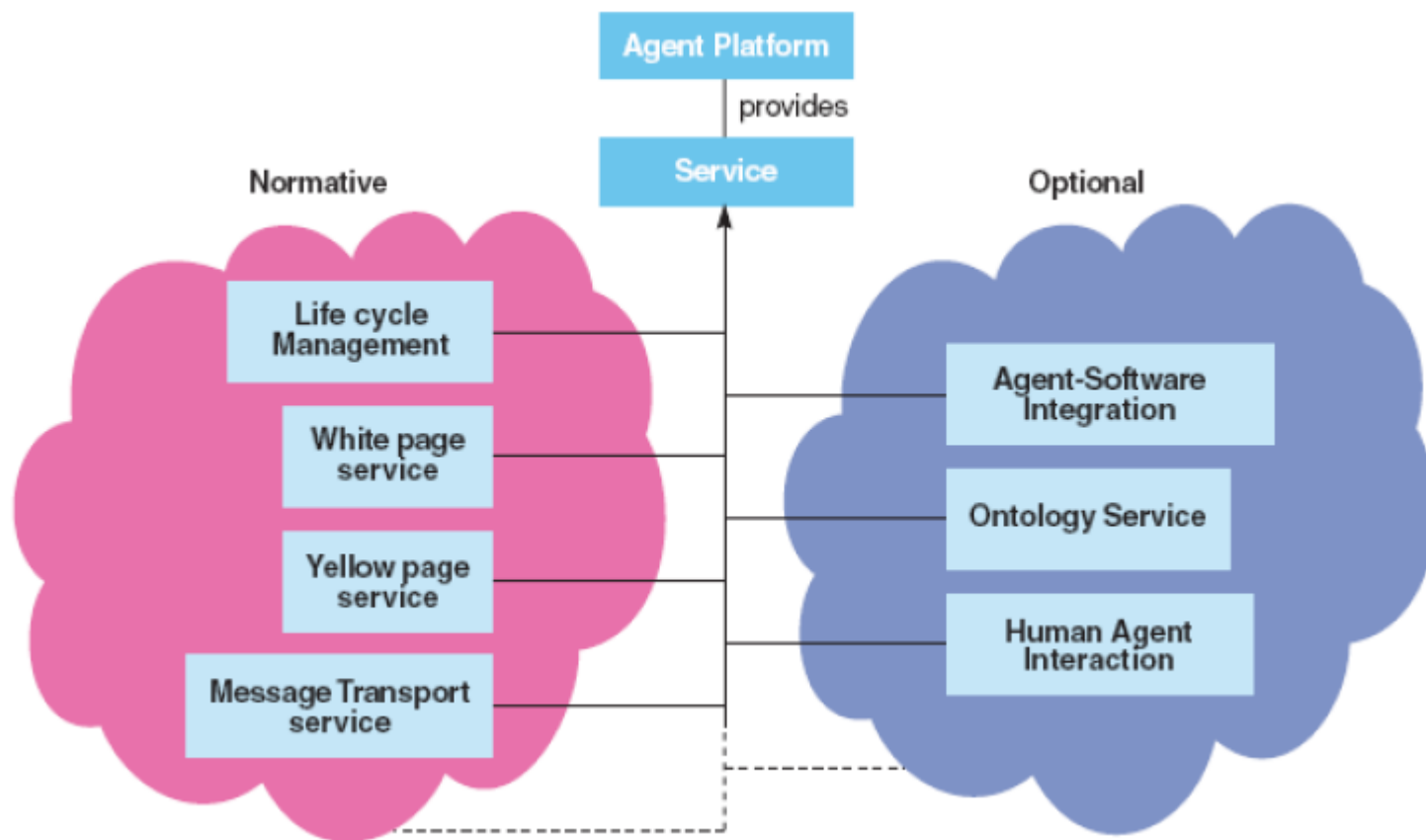
Кодиране

Транспортен

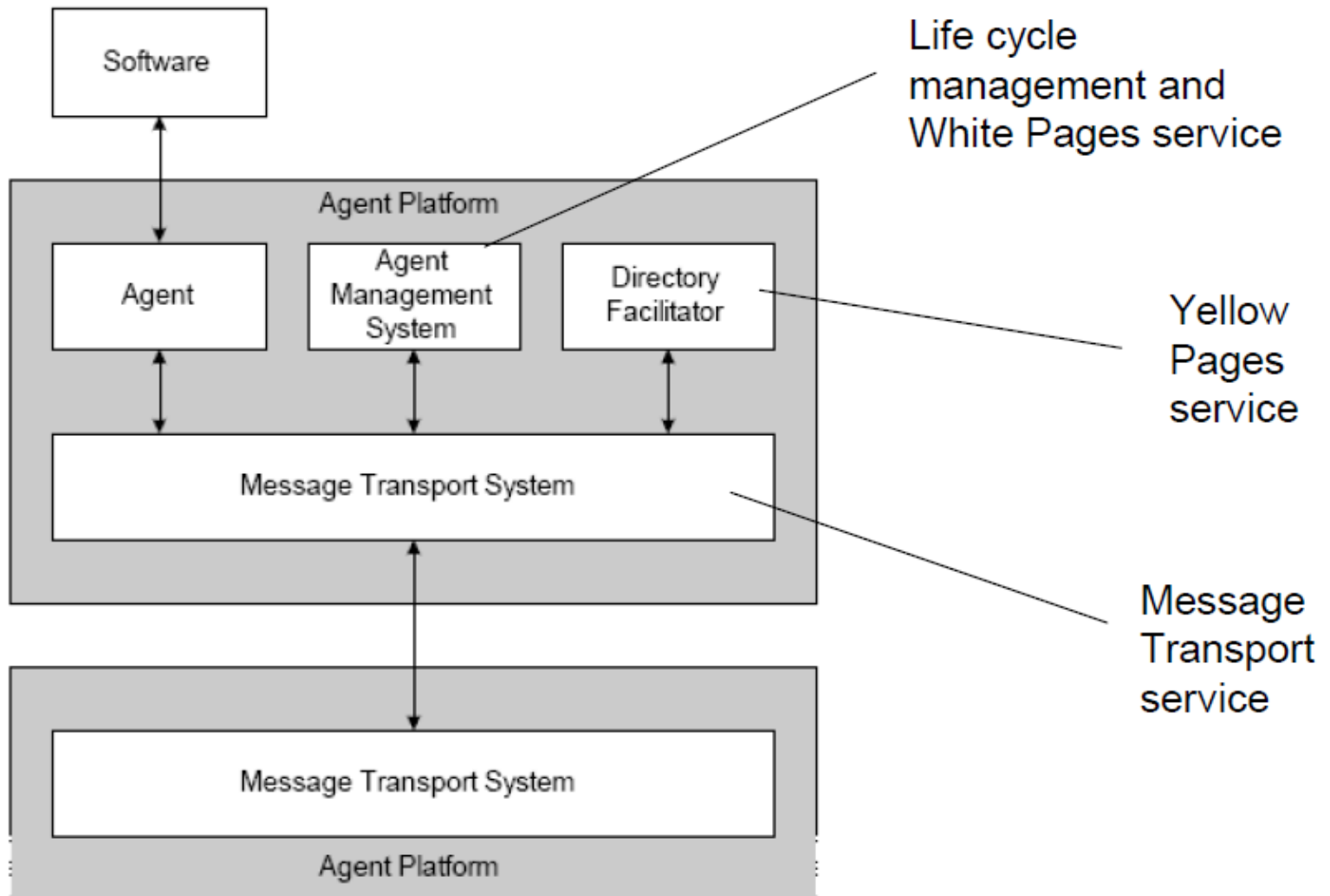
- XML, String, Bit-Efficient
- IIOP, WAP, HTTP

FIPA архитектура: Услуги

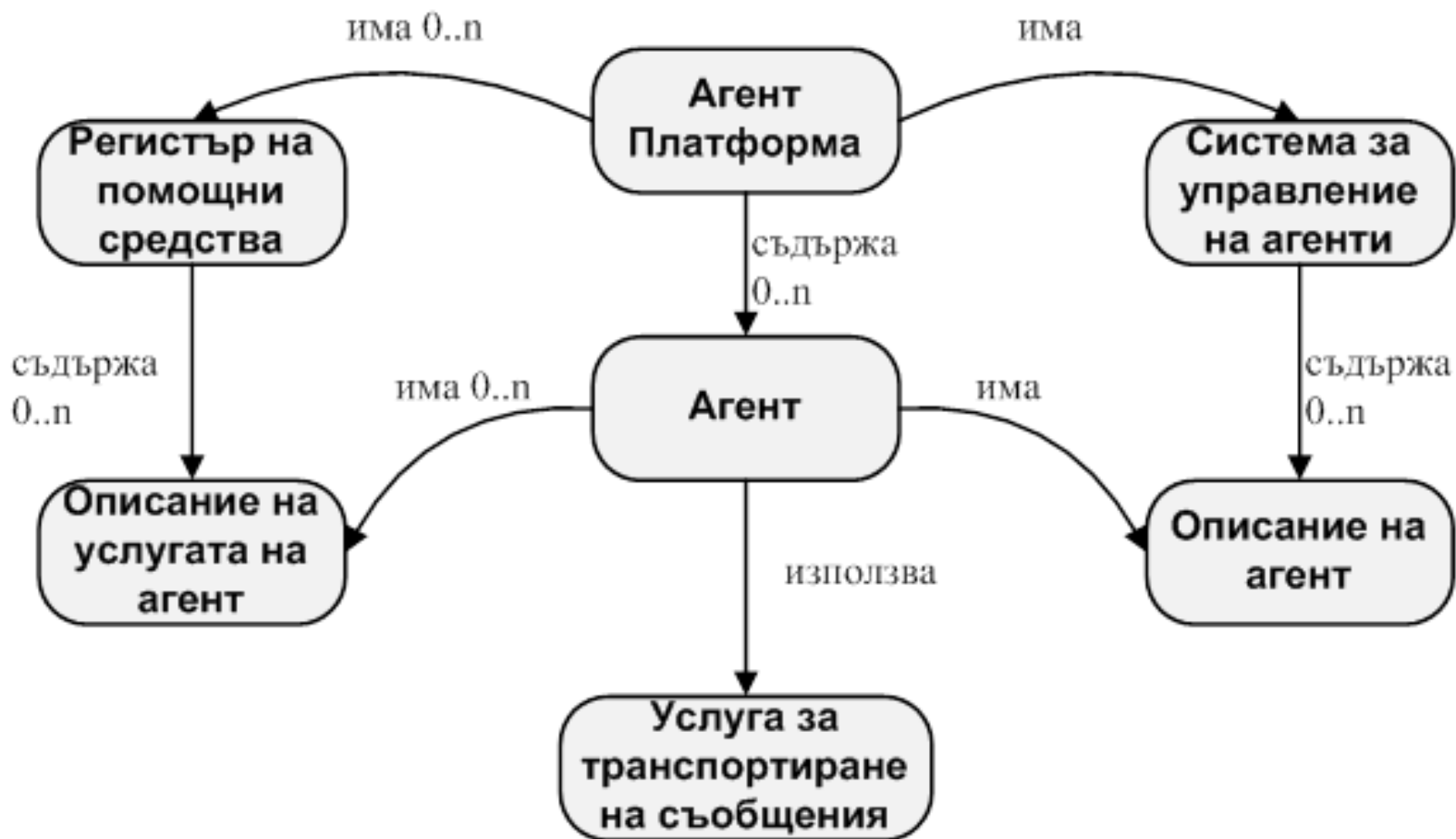
- Агентите се хостват в *агент-базирани платформи*.
- Услугите се предоставят от АБП:



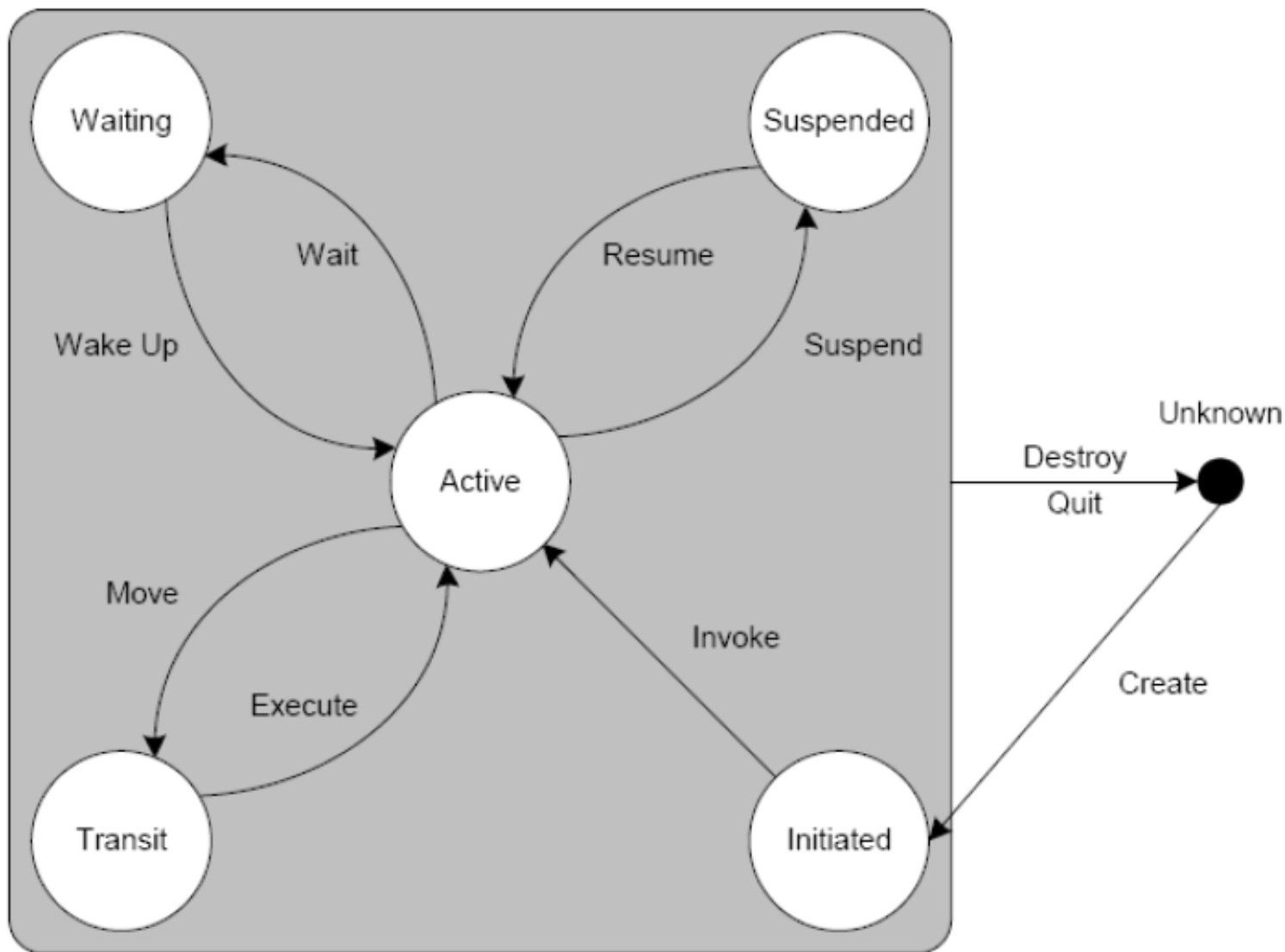
FIPA архитектура: елементи на системата



Онтология на модела за управление на агенти



Жизнен цикъл на агент



Жизнен цикъл на агент – 1/2

- **INITIATED**: обектът агент е създаден, но не е регистриран в AMS, няма име и адрес и не може да комуникира с другите агенти.
- **ACTIVE**: обектът агент е регистриран в AMS, има редовно име и адрес и може да получи достъп до всички услуги в платформата.
- **SUSPENDED**: обектът агент е текущо спрян. Поведението на агента не се изпълнява.

Жизнен цикъл на агент – 2/2

- **WAITING**: обектът агент е блокиран и е в състояние на изчакване докато някакво условие не се изпълни (обикновено когато пристигне съобщение).
- **DELETED**: обектът агент е унищожен. Изпълнението му е приключил и вече не е регистриран в AMS.
- **TRANSIT**: при мобилните агенти, които преминават в това състояние когато мигрират към ново местоположение. Системата поддържа буфер от съобщения, които се изпращат към новото нахождение.

Agent Management System (AMS) – 1/2

- AMS е задължителен елемент на АП и само един AMS ще продължат да съществуват в един АП.
- В AMS представлява управляващият орган на АП и ако АП е разпределена на няколко машини, тогава AMS е управляващ орган за всички машини.
- AMS е отговорен за управлението на работата на АП, като например създаване и унищожаване на агенти, контрол по миграцията на агенти до и от АП.
- В AMS се поддържат индекси на всички агенти, които в момента пребивават в АП, които включват AID (идентификатор на агентите).

Agent Management System (AMS) – 2/2

- Осигурява основните функции по регистриране, отрегистриране, промяна, търсене (по име), описание.
- Може да възложи на АП да извършват следните дейности: да спре агент, да унищожи агент, да създаде агент, да поднови агент за изпълнение, управление на ресурсите и др.

Directory Facilitator (DF)

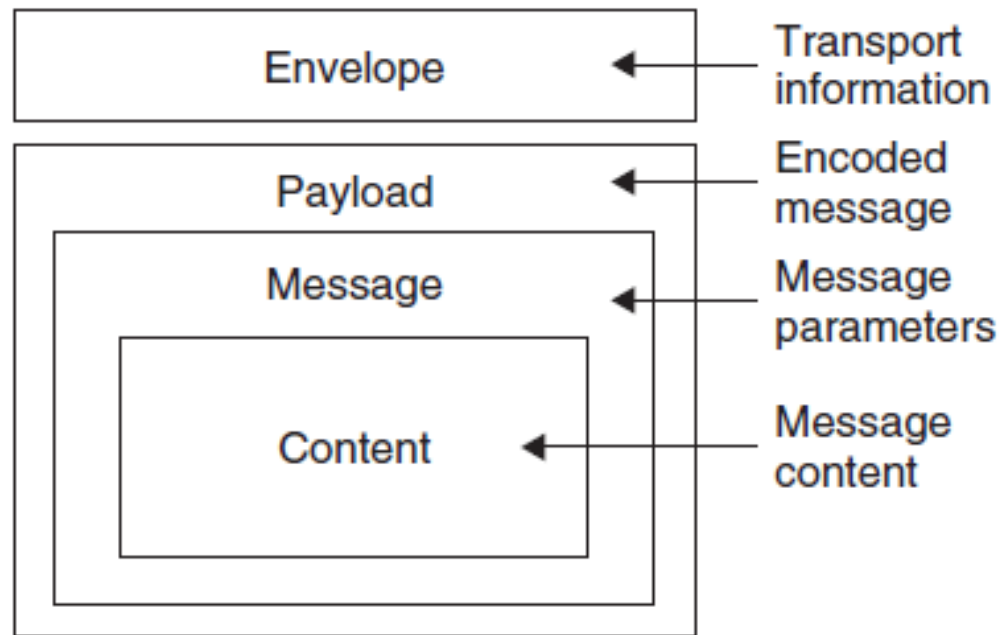
- DF е компонент на АП, която осигурява услугата “жълти страници” за агентите.
- Осигурява основните функции: *register*, *deregister*, *modify*, и *search*.
- Когато се използва агент-базирана платформа като JADE, операциите с DF обикновено се извършват с помощта на API функции.

```
(  
  (action  
    (agent-identifier  
      :name df@foo.com  
      :addresses (sequence iiop://foo.com/acc)  
    )  
  (search  
    (df-agent-description  
      :ontologies (set meeting-scheduler)  
      :languages (set fipa-sl0 kif)  
      :services (set  
        (service-description  
          :name profiling  
          :type meeting-scheduler-service)  
        )  
      (search-constraints :max-depth 2)))  
  )  
)
```

Message Transport Service (MTS)

- MTS доставя съобщения между агенти в рамките на АП и на агентите, които са в други АП.
- Всички FIPA агенти имат достъп до най-малко една MTS.
- Agent Communication Channel (ACC) предоставя услуга пряко на агентите на АП.
- MTP се използва за извършване на физическото пренасяне на съобщения между два ACC.
 - над Internet Inter-Orb Protocol (IIOP)
 - над HTTP

FIPA съобщение



Агент-базирани платформи

- FIPA спецификации:
(<http://www.fipa.org/resources/livesystems.html>):
- Платформи:
 - JADE
 - Cougaar
 - AgentFactory (partially FIPA-compliant)
 - 3APL platform
 - Jason (AgentSpeak APL)
 - UBIWARE Platform (on top of JADE) with S-APL

ИЗТОЧНИК

- **Design of Agent-Based Systems (TIES433)**

Artem Katasonov, University of Jyväskylä