

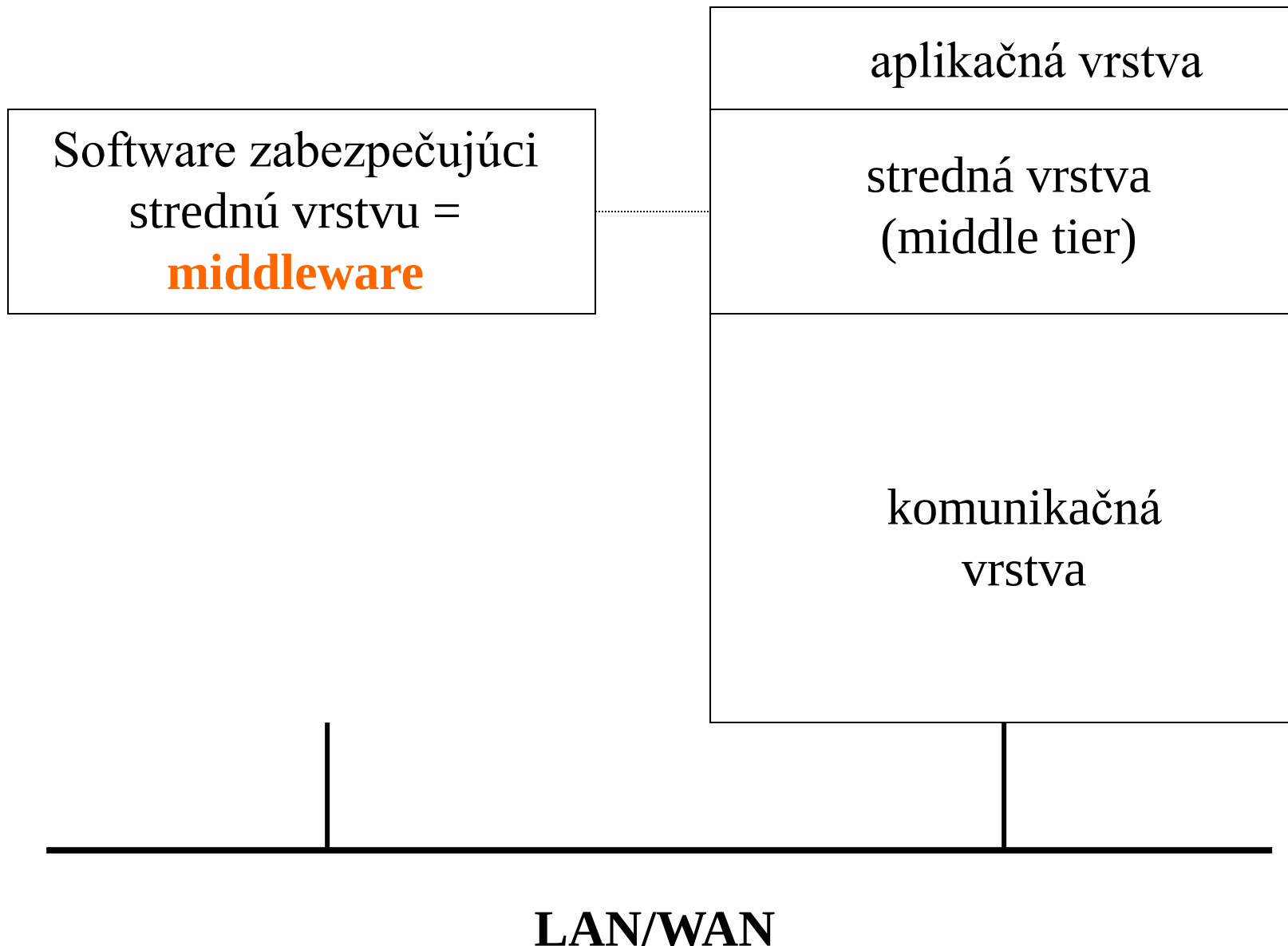
Multiagentové systémy

Andrej Lúčny

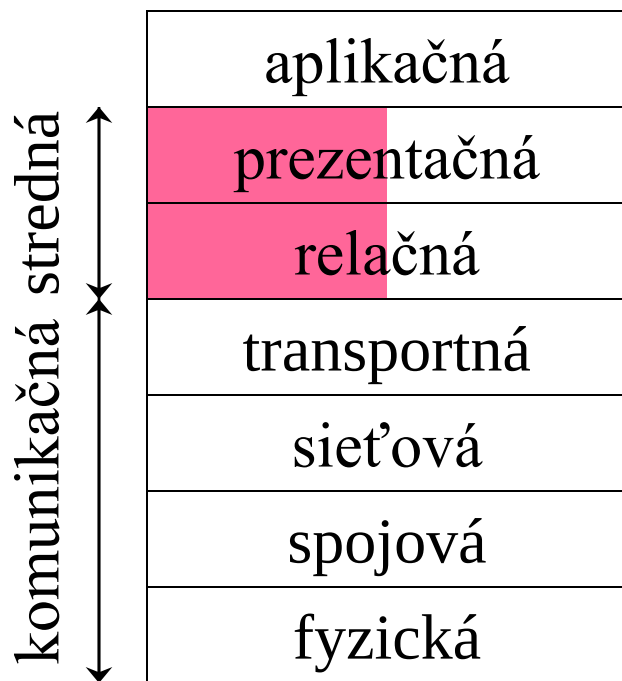
KAI FMFI UK

lucny@fmph.uniba.sk

<http://www.agentspace.org/mas>



MAS ako middleware



- **software podporujúci tvorbu MAS je vždy postavený na nejakej komunikačnej základni**
- **predstavuje špecifický druh middleware založeného na message passing-u**
- **môže sa opierať o iný middleware** (často je realizovaný pomocou distribuovaných objektov)

MAS API

- Aké rozhranie poskytuje implementácia MAS pre aplikačnú úroveň ?

Sú dve možnosti

- Prevládajúca priama komunikácia (peer-to-peer) (napr. JADE)
- Prevládajúca nepriama komunikácia (stigmergic communication) (napr. Cougaar)

MAS
s prevládajúcou
priamou
komunikáciou

Služby priamej komunikácie

Základnou službou je

- asynchrónne poslanie správy t.j. SEND
t.j. vyvolanie vhodnej metódy s vhodnými parametrami na strane sendera
vo všeobecnosti táto môže priamo zavolať aplikačný kód prijímateľa (callback, typický pre aktory) alebo uložiť správu do nejakej zdieľanej pamäti odkiaľ si ju vlákno prijímateľa vyčíta (agentový prístup). Agent teda potrebuje zavolať nejakú metódu na prijatie, RECEIVE



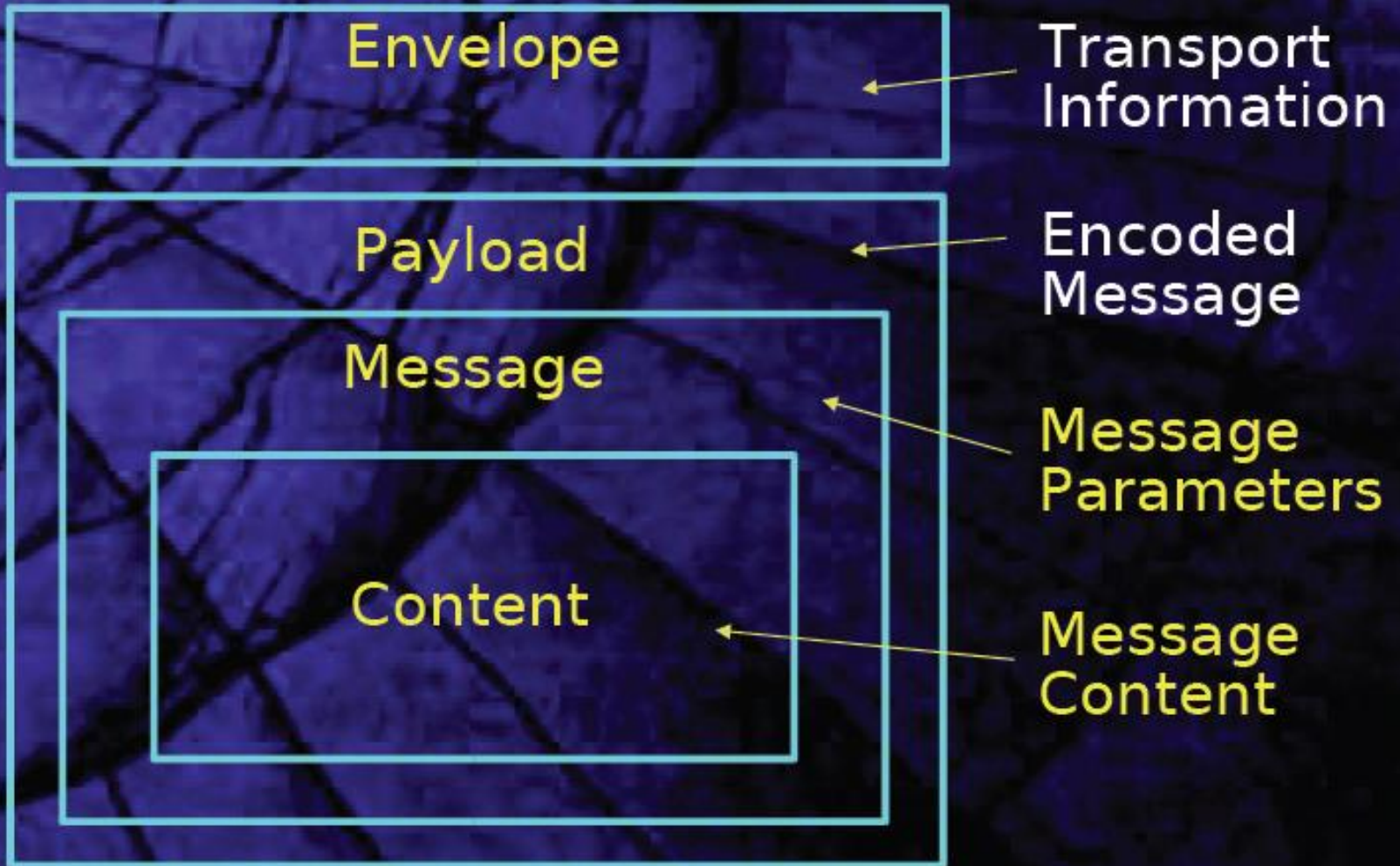
- Java Agent DEvelopment Framework
- Middleware postavený nad RMI
- rieši
 - komunikáciu medzi agentami (ACL) a jej nadväzovanie (AMS, DF)
 - Súbežný beh agentov (Jade container)
- Implementácia FIPA 97 v Java

FIPA 97

- Started work in 1997
- At peak comprised 60 members
- Primary specifications became standard in 2002
- Work ongoing in Modeling, Methodology, Semantics & Services

ACL = Agent Communication Language

- FIPA Message Structure -



Atribúty performatívov ACL

- ACL Message Structure -

FIPA ACL Message Elements

<i>Element</i>	<i>Description</i>
performative	What action the message performs
sender	Initiator of the message
receiver	Recipient of the message
reply-to	Recipient of the message reply
content	Content of the message
language	Language used to express content
encoding	Encoding used for content
ontology	Ontology context for content
protocol	Protocol message belongs to
conversation-id	Conversation message belongs to
reply-with	Reply with this expression
in-reply-to	Action to which this is a reply
reply-by	Time to receive reply by

Príklad správy v ACL (a FIPA-SL)

- ACL Message Example -

```
(request
  :sender (:name dominic-agent@whitestein.com:8080)
  :receiver (:name rex-hotel@tcp://hotelrex.com:6600)
  :ontology personal-travel-assistant
  :language FIPA-SL
  :protocol fipa-request
  :content
    (action movenpick-hotel@tcp://movenpick.com:6600
      (book-hotel (:arrival 25/11/2000) (:departure 05/12/2000) ...
    ))
)
```

Príklad správy v Java Serialization

(INFORM

:sender (agent-identifier :name Writer@10.102.101.216:1099/JADE)

:receiver (set (agent-identifier :name Reader@10.102.101.216:1099/JADE))

:X-JADE-Encoding Base64

:content

"rO0ABXNyAAINeU1lc3NhZ2X3XJJaURjcRQIAAUwAB2NvbnRlbnR0ABJMamF2
YS9sYW5nL1N0cmlyZzt4cHQACkFob2ogSmVsa2E="

:language JavaSerialization)

Príklad správy v XML

(INFORM

:sender (agent-identifier :name Writer@192.168.1.15:1099/JADE)

:receiver (set (agent-identifier :name Reader@192.168.1.15:1099/JADE))

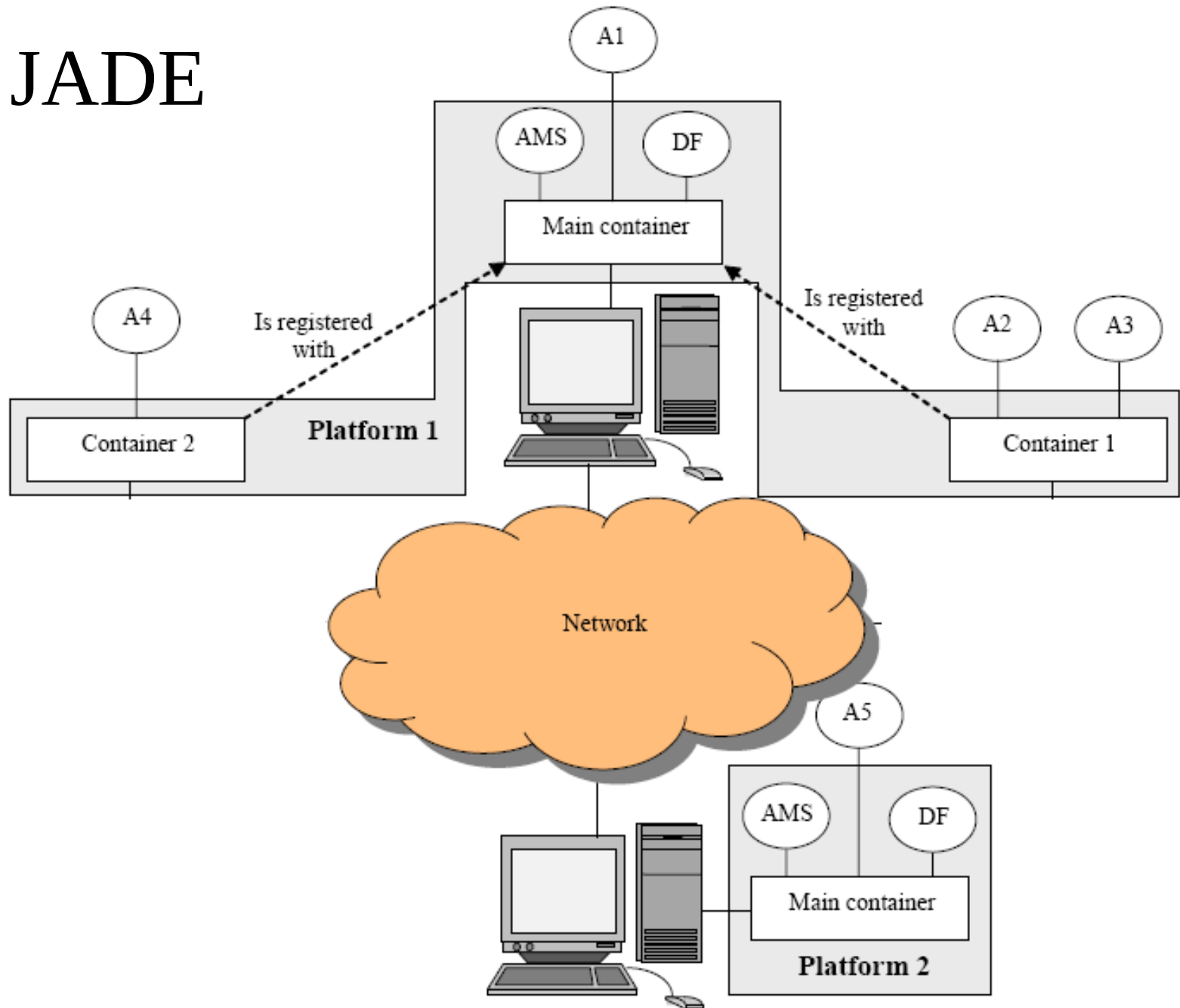
:content "<primitive type=\"FLOAT\" value=\"3.1415927F\"/>"

:language XML :ontology FIPA-Agent-Management)

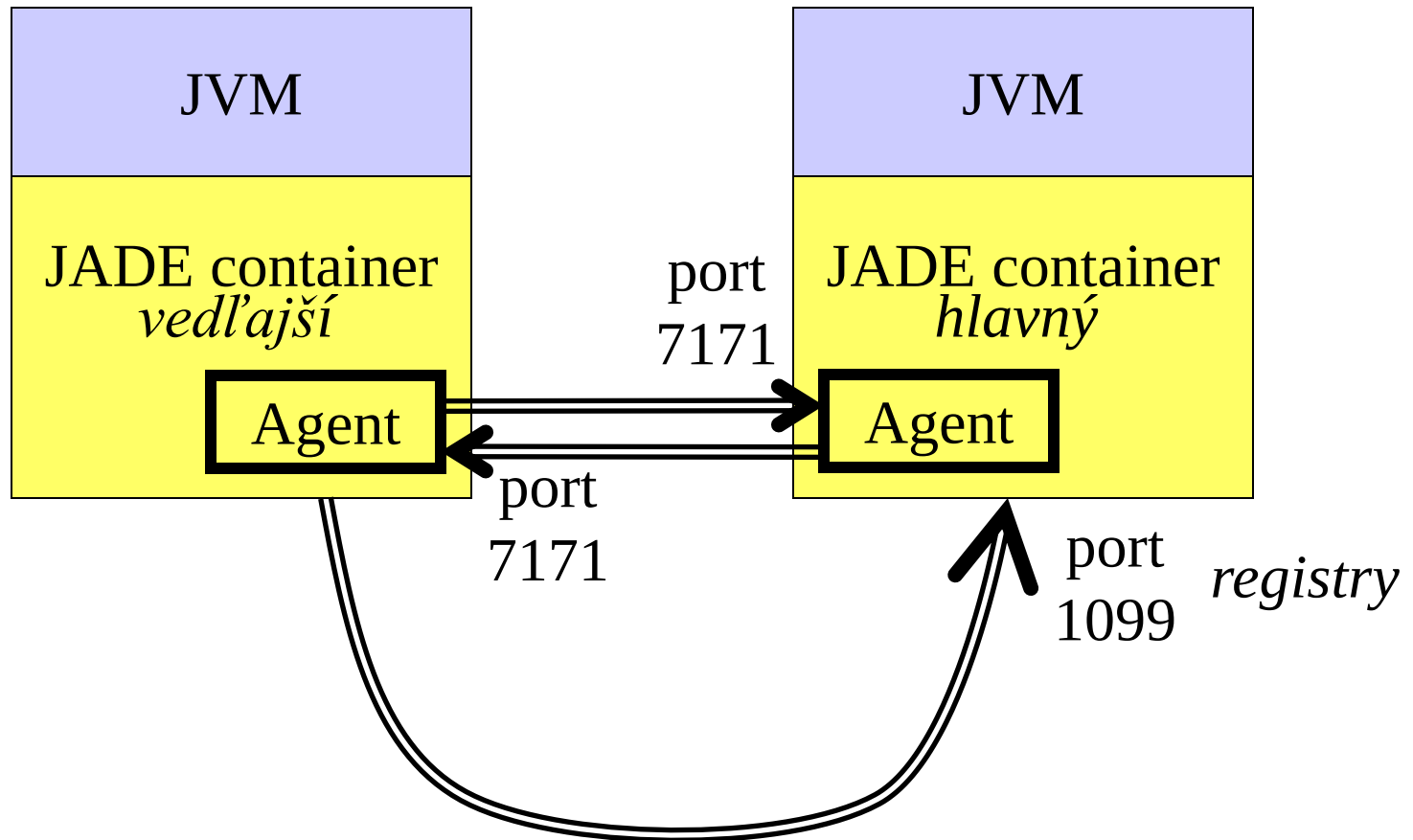
Content v ACL (Marshalling)

- FIPA-SL, FIPA-KIF
- Serializované objekty Javy (RMI)
- ACL/XML
- ACL/Bit-efficient

JADE



TCP sockety pre JADE



Jade container

- Asynchrónny model agentov
- Agent má vlastné riadenie, pričom je potenciálne multivláknové
- Container zakrýva rozdiely v sieťovej a lokálnej komunikácii (obsahuje a skrýva všetky potrebné stub-y)

Jade container

- Čo container, to jedna JVM
- Môže ich byť viac na jednom uzle, ale vtedy práve jeden z nich musí byť hlavný (a ten obsadzuje port 1099)
- Container sa môže napojiť na hlavný container na inom uzle a takto spôsobiť, že sa agenti na rôznych uzloch vidia.

Jade: Agent

- Agent beží len v containeri, ktorý realizuje jeho rozhranie
- Agent má metódu **setup()**
- V nej môže vo svojom vlákne realizovať svoju činnosť, ale častejšie je, že inicializuje rôzne správania metódou **addBehaviour()**

Jade: Behaviour

- SimpleBehavior
- ParallelBehavior
- CyclicBehaviour
- TickerBehaviour
- OneShotBehaviour
- ReceiveBehaviour
- WakerBehaviour

Jade: Behaviour – příklady

```
protected void setup() {  
    ...  
    addBehaviour(  
        new CyclicBehaviour () {  
            public void action () {  
                ...  
            }  
        }  
    );  
}
```

```
protected void setup() {  
    ...  
    addBehaviour(  
        new MsgReceiver (this, MessageTemplate.MatchAll(), Long.MAX_VALUE,  
            new DataStore(), "ObjectReaderAgent") {  
            protected void handleMessage (ACLMessage msg) {  
                ...  
            }  
        }  
    );  
}
```

Jade Agent Management System

- Každý agent má svoje AID a meno a obýva určitý container
- Aby bol viditeľný pre ostatných môže zaregistrovať a odregistrovať tzv. service
- Môže zisťovať AID iných agentov podľa mena servisu, ktorý si zaregistrovali (yellow pages)
- Môže zisťovať AID agentov podľa miesta kde sa nachádzajú (white pages)
- Môže používať ACL na vzájomnú komunikáciu s agentami ktorých AID vie

Jade Directory Facilitator

- Slúži na registráciu servisov
- Dokáže servisy organizovať v tzv. adresároch, čo je virtuálna štruktúra pre všetky container-y ktoré sa vidia.

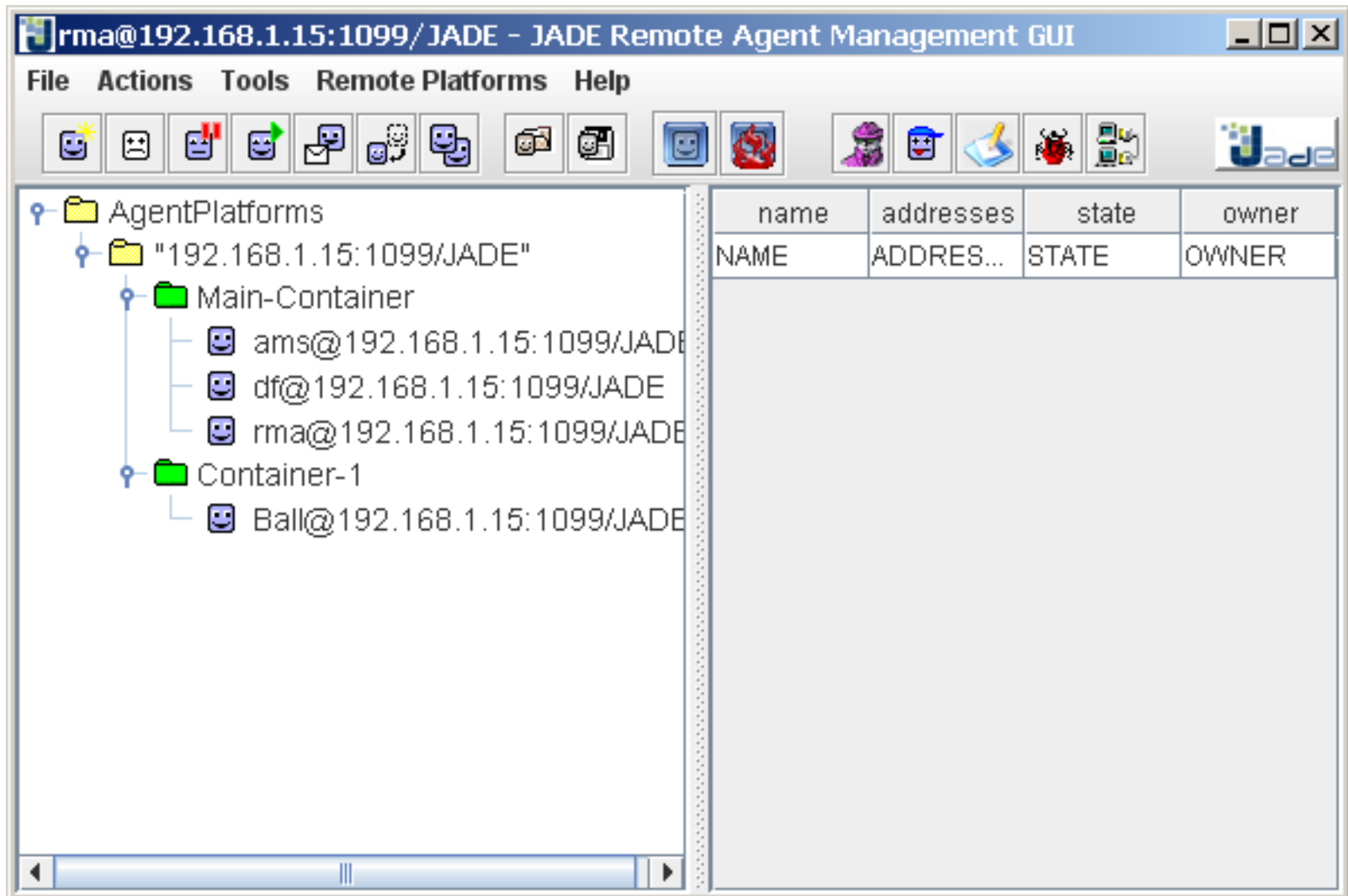
Registração

```
DFAgentDescription dfd = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType(„ServiceName");
sd.setName(getName());
dfd.addServices(sd);
dfd.setName(getAID());
try {
    DFService.register(this,dfd);
} catch (FIPAException e) {
}
```

Vyhľadanie agenta podľa servisu

```
AID reader;
DFAgentDescription dfd = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType("ServiceName");
dfd.addServices(sd);
try {
    for (;;) {
        DFAgentDescription[] result = DFService.search(this,dfd);
        if ((result != null) && (result.length > 0)) {
            dfd = result[0];
            reader = dfd.getName();
            break;
        }
        Thread.sleep(1000);
    }
} catch (Exception e) {
}
```

Jade GUI



Ďalšie info

- <http://jade.tilab.com/>
- <http://jade.tilab.com/download/add-ons/>
- <http://www.iro.umontreal.ca/~vaucher/Agents/Jade/JadePrimer.html>

MAS s priamou komunikáciou je distribuovaným systémom typu:

Peer-to-peer

- minimálna potreba centrálnych serverov - opiera sa len o nejakú client-server službu ako je naming service
- Každý uzol je serverom pre ostatných peerov a zároveň klientom ostatných peerov

