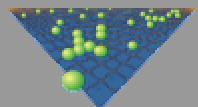




Overview

- Agent platform announcement + DF federation
- Multiple DFs for different technologies
- DF plug-ins for discovery protocols
- Service Discovery Channel (SDC)
- Discovery Agent (DA, Jamie Lawrence like)





Agent platform announcement + DF federation

Adhoc management
service agent (AMSA)

AMSA functionality (see whitepaper)

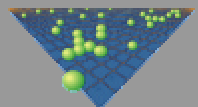
- periodically announce agent platform via HTTPMU (through ACC)
- federate DFs of found remote agent platforms
- lease management for automatic defederation

ACC

HTTPMU

HTTP

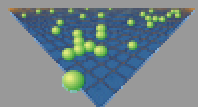
...





Agent platform announcement + DF federation (2)

- Advantages
 - Minimal additions to existing FIPA standards
 - Adhoc agent discovery through existing mechanisms (DF)
 - Transparent mapping of any service nearby to the DF
- Disadvantages
 - No usage of built-in service discovery mechanisms of existing adhoc technologies (e.g. Bluetooth)
 - Efficient usage of service description distribution with peer-to-peer technologies (e.g. future JXTA implementations, Chord) also not possible (lack of different service types)
 - DF required on each platform

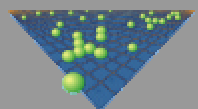
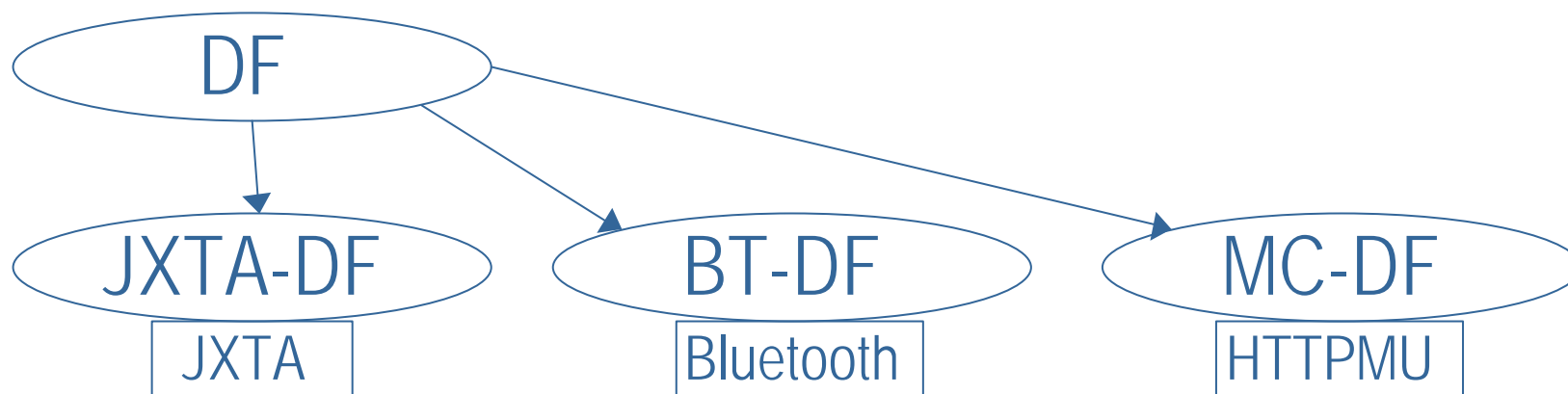




Multiple DFs for different technologies

Functionality

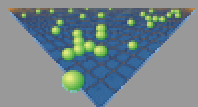
- one DF for every adhoc/p2p technology
- DF federates with any of these specialised DFs
- Agents may register at any or all of these DFs
- Searches can be directed to only some adhoc/p2p discoveries
- New "Subscribe" action for DFs





Multiple DFs for different technologies (2)

- Advantages
 - Overall architecture very "clean"
 - Transparent mapping of any service nearby to the top level DF (by federating DFs)
 - Implementation allows subclassing (no code replication)
- Disadvantages
 - Multiple DF instances - may be very memory intensive
 - Subscribe action required - what happens with subscribe at federated DFs?





DF plug-ins for discovery protocols

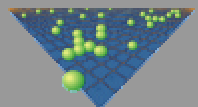
Functionality

- registration at DF automatically announces agent in adhoc networks
- search at DF searches adhoc networks also
- new method "subscribe" required at DF for eventing
- selection where to search and where to register agents through additional search/subscribe/registration parameters

DF

Service Discovery Interface

Bluetooth	JXTA	HTTPMU	LDAP	...
-----------	------	--------	------	-----

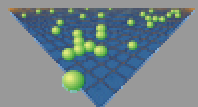


Information &
Communications
Intelligent
Autonomous Systems



DF plug-ins for discovery protocols (2)

- Advantages
 - DF gets extended to adhoc networks
 - Fits seamlessly to FIPA standards: DF remains as the one and only directory (from an agent's view)
 - Transparent mapping of any service nearby to the DF
- Disadvantages
 - Additional parameters necessary
 - Additional DF action "subscribe" necessary
 - Full DF required - DF gets even larger



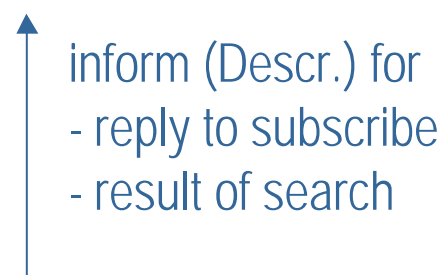


Service Discovery Channel (SDC)



SDC Description:

- minimally contains AID and service types
- alternatively full DF agent description



SDC

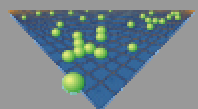
JXTA

BT

HTTPMU

LDAP

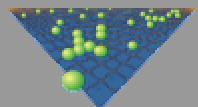
...





Service Discovery Channel (SDC) (2)

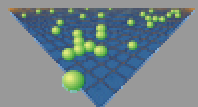
- Same level as ACC/MTS
- Does not have to be an agent; e.g. access through method calls possible
- Agent/service descriptions including service type
- Distinction between local and remote discovery
 - SDC only able to discover remote agents
 - If local search required DF still necessary
- Filtering of results in the discovery layer of the underlying protocols (Bluetooth, JXTA, ...)





Service Discovery Channel (SDC) (3)

- **Advantages**
 - Small footprint (no ACL messages necessary to communicate with SDC)
 - Efficient registration/lookup in Peer-to-Peer- (e.g. JXTA) and Ad-hoc-networks (e.g. Bluetooth) because of availability of service type (filtering)
 - Search/subscribe for DFs make DF federations possible
 - DF is optional
 - No existing FIPA entities have to be modified
- **Disadvantages**
 - May not fit seamlessly into current FIPA infrastructure -> additional communication layer
 - Explicit distinction of local and remote lookup mechanisms (DF vs. SDC)

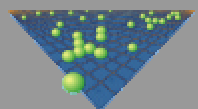
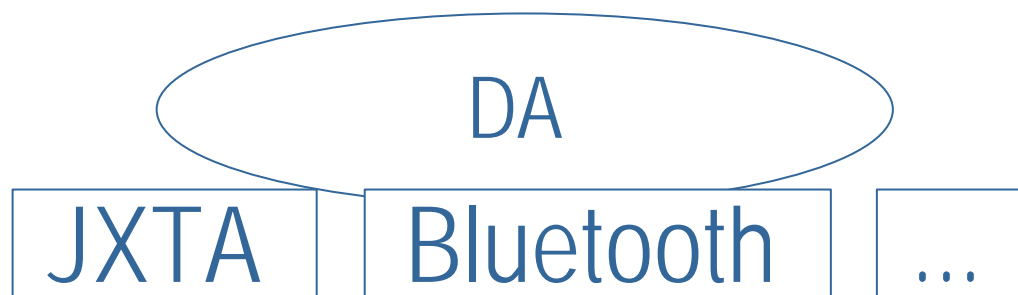




Discovery Agent (DA, based on Jamie Lawrence)

Functionality

- Similar to SDC, but as an agent
- No descriptions: Subscribe returns any agent that comes in range
- DA registers any agent registered locally at DFs coming in range
- Leasing at DF
- DF optional





Discovery Agent (DA, Jamie Lawrence like) (2)

- Advantages
 - DA is an agent -> fits to FIPA standards
 - DF is optional
- Disadvantages
 - DA communicates by low level protocols (e.g. JXTA) without going through ACC/MTS -> problem with FIPA architecture
 - DA provides DF like functionality -> almost same memory requirements
 - Scalability problem when a lot of agent platforms with a DF come in range
 - DA fires event for any agent coming in range -> no filtering for specific agent type in low level protocols possible

