FIPA-OS Feature Overview

Agent Technology Group

Nortel Networks

February 2000

FIPA-OS - Aims

FIPA-OS is a Open Source implementation of FIPA and is available for free.

http://www.nortelnetworks.com/fipa-os for more information.

- **A ‘reference implementation’ of the FIPA open standard for agent interoperability**
- **OS means Open Source, freely available and modifiable source code (cf Linux)**
- **Enables adoption of FIPA without the need to implement the specifications**
- **Assist in validating and evolving FIPA standards**

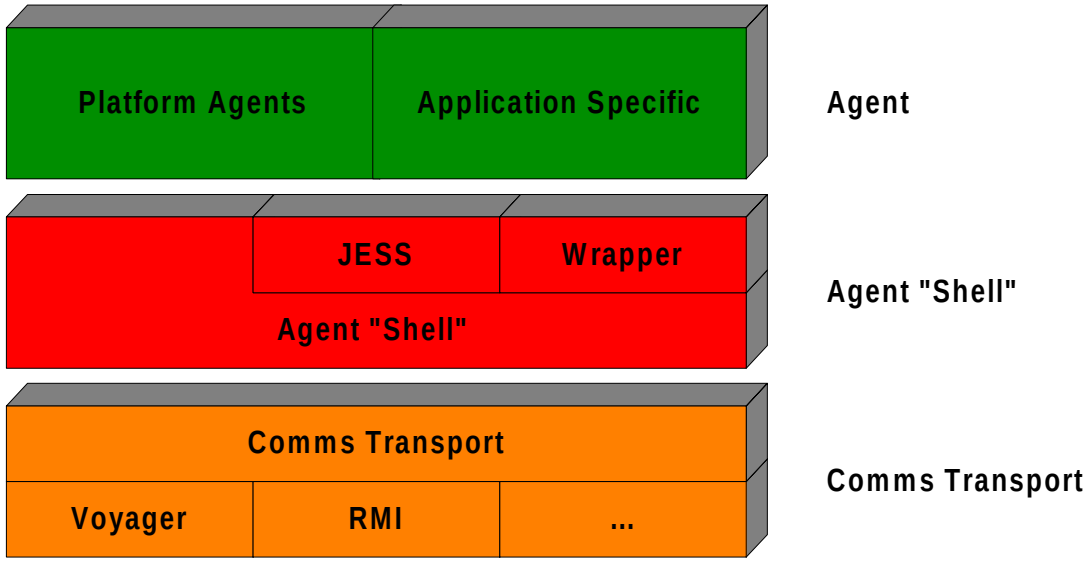
FIPA-OS - Features Summary

- **Platform Agents**
  - AMS, DF, (ACC)
- **Agent Shell**
  - ACL, SL0 and XML\RDF (RDF encoding of SL0) parsers
  - Persistence abstract interface (bindings for serialisation)
  - Transport abstract interface (bindings for Voyager and SunIDL
- **Configuration**
  - XML\RDF Platform and Agent profiles
  - IOR distribution via HTTP

FACTS

FIPA-OS - Layered Model

- **FIPA-OS can be logically viewed as 3 layers, each composed on plugable components**



FACTS

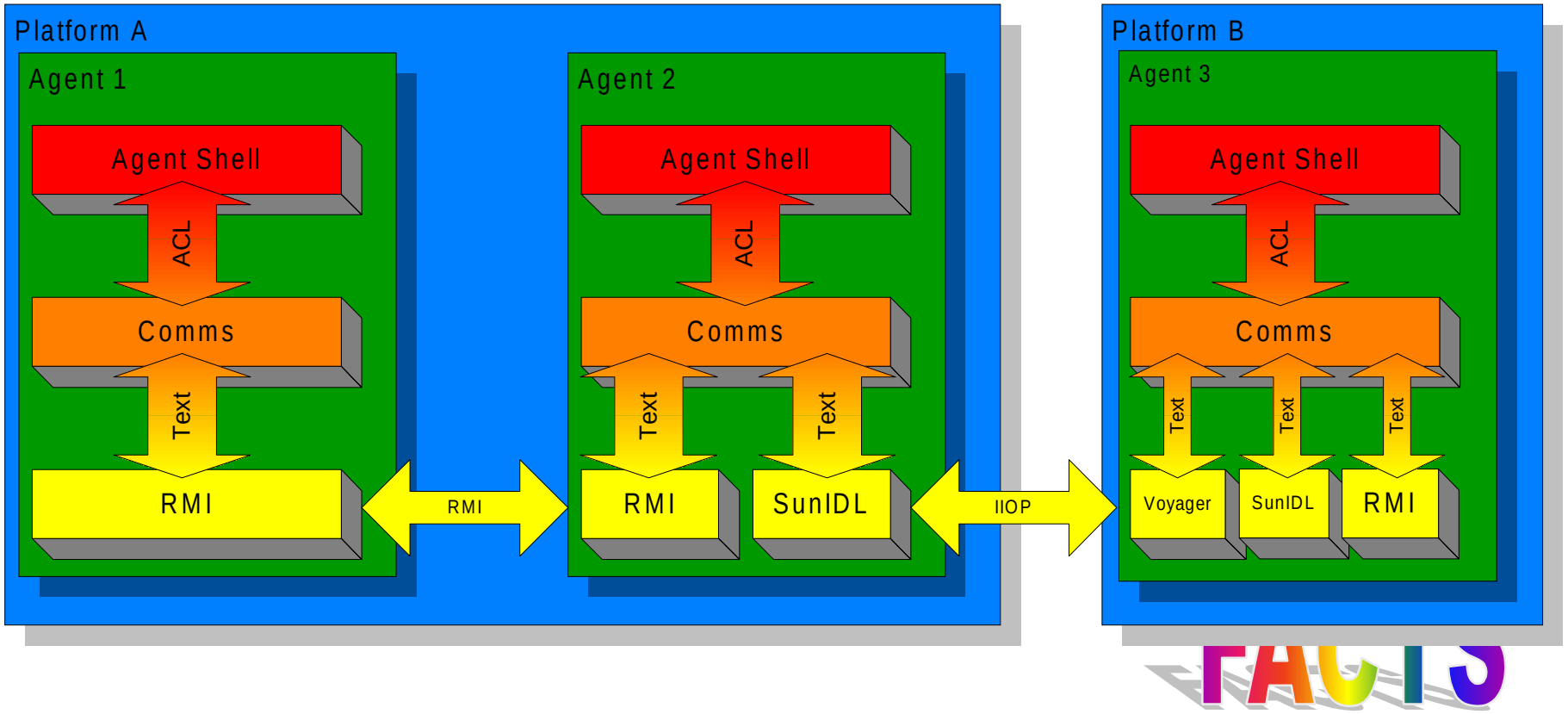
Comms Transport (1/2)

- **Support for multiple transport means**
  - IIOP using Java ORB & Voyager ORB (FIPA97 and FIPA99 compliant), or RMI
  - Several transports can be in use at any time
- **Support for multiple ACL encodings**
  - FIPA ACL
  - ACL DTD
- **Support for multiple content language encodings**
  - FIPA-SL0/1
  - FIPA-RDF

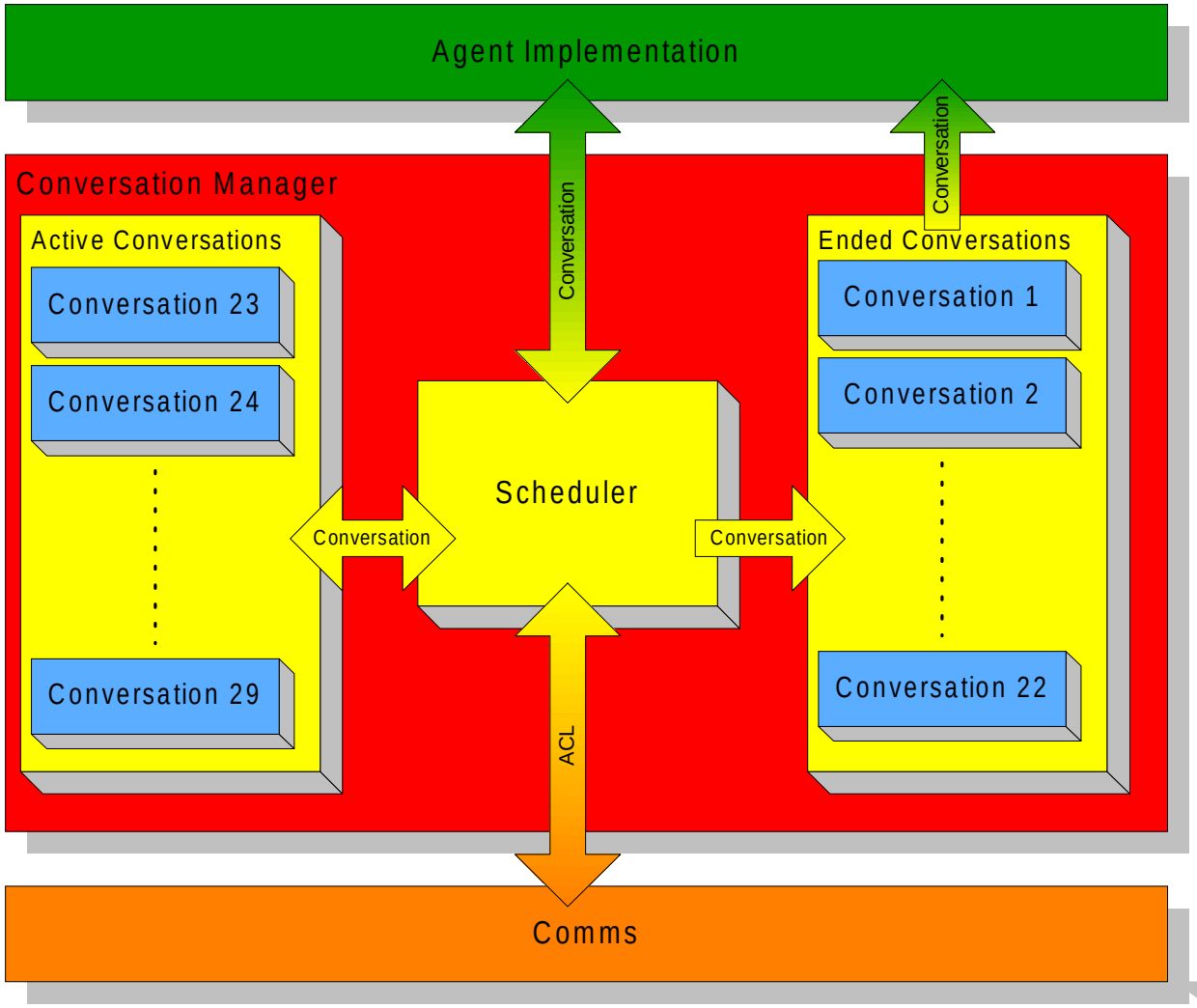
FACTS

Comms Transport (2/2)

- **Example scenario**
  - RMI used for intra-platform communications
  - IIOP using SunIDL/Voyager for inter-platform communication



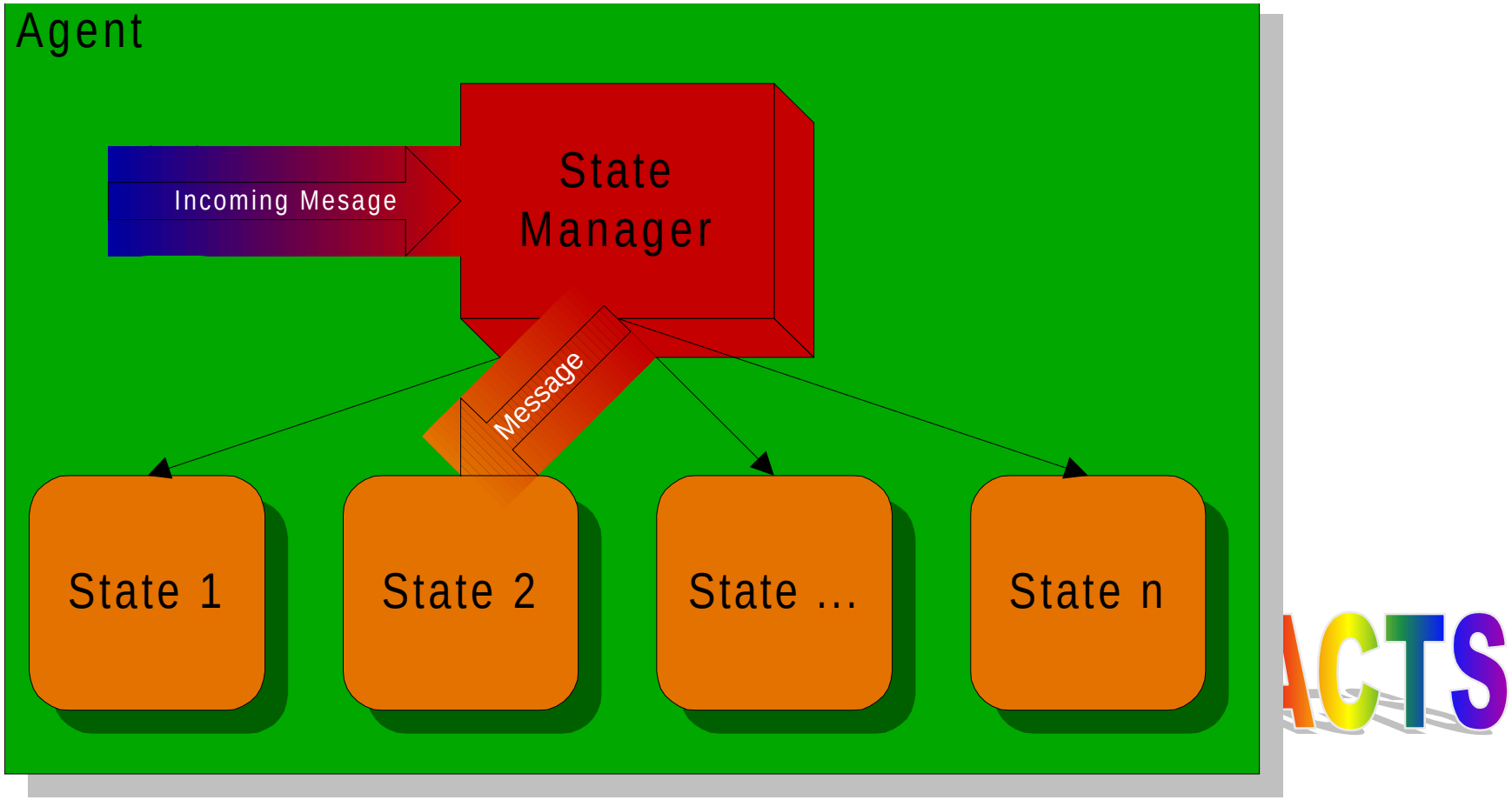
Conversation Manager



FACTS

State Manager

- Separates agent ‘tasks’ into distinct objects
- Messages are automatically routed to the correct state



Agent “Shell” - Java & Wrapper

- **Agent “Shell”**
  - Provides a basic framework for constructing agents
  - Messaging and state components available by default
  - Skill components can be added to tailor the agent
- **FIPA Wrapper Agent “Shell”**
  - Extends the basic agent shell by adding FIPA compliant wrapper functionality
  - Used to develop agents to control non-agent (legacy) software

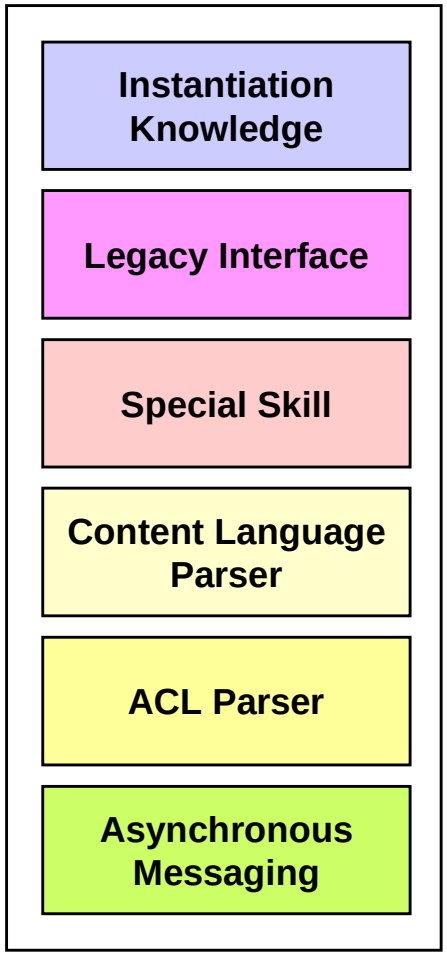
Agent “Shell” - JESS Agent

- **JESS Agent Conforms to FIPA conversations**
  - Maps performatives onto basic Jess operations
    I.e. infrom(FIPA)=assert(JESS)
  - If a conversation requires response from the KB then the KB will respond
    e.g. a FIPA request has an inform in it. The content of the inform comes from KB
- **The interface to Jess uses content objects**
  - If the parser exists the content language can be used
    e.g. FIPA-RDF

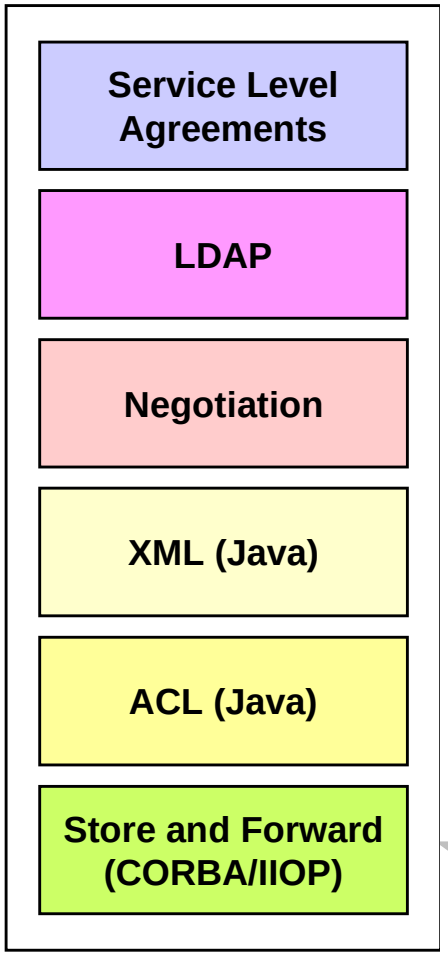
FACTS

FIPA-OS Agent "Shell"

Agent Shell (Abstract)

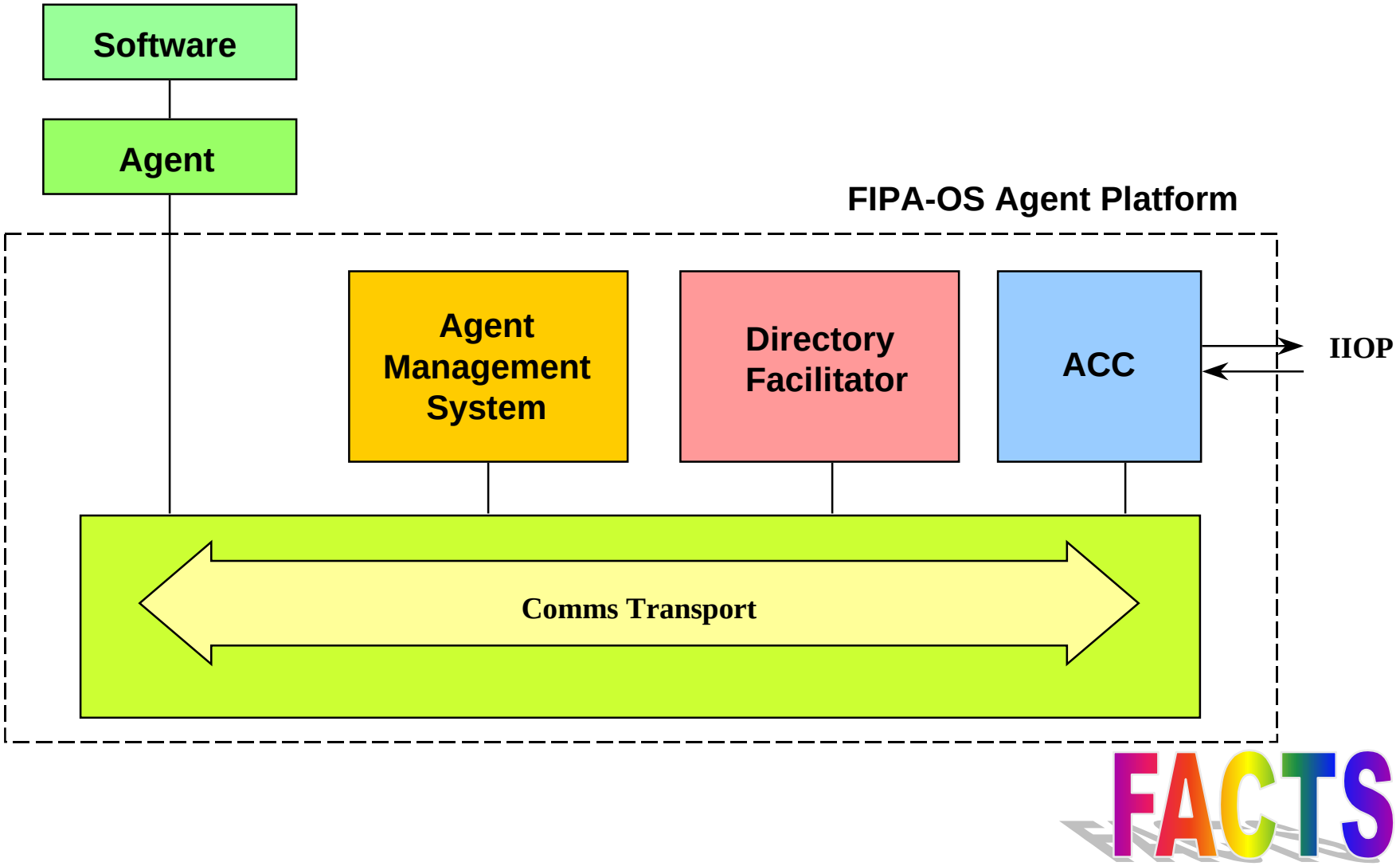


Example agent Instance



FACTS

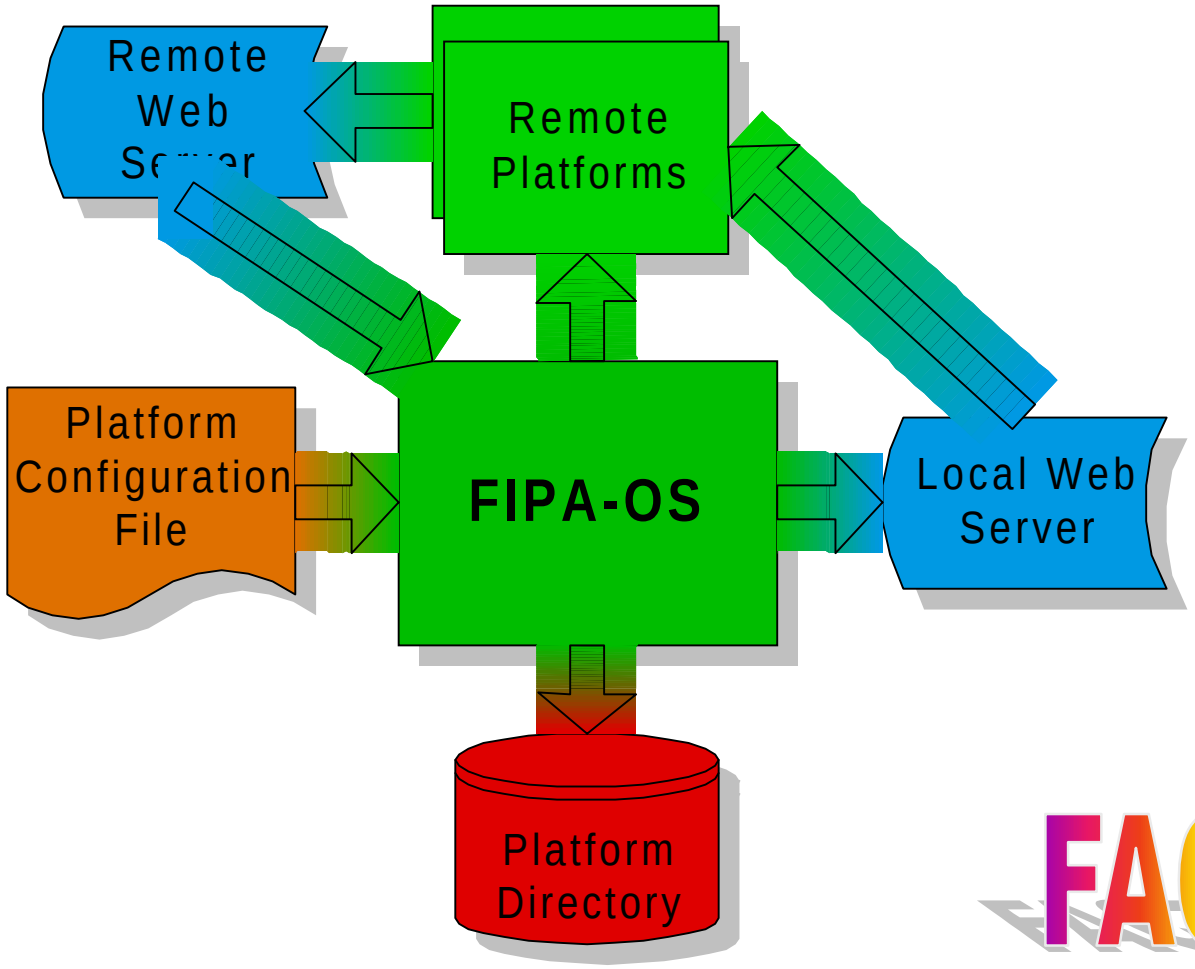
FIPA-OS - Platform Agents



FACTS

Remote Platform Bootstrapping

- **Platform IORs published to web servers**
- **Dynamically contact remote platforms**



FACTS

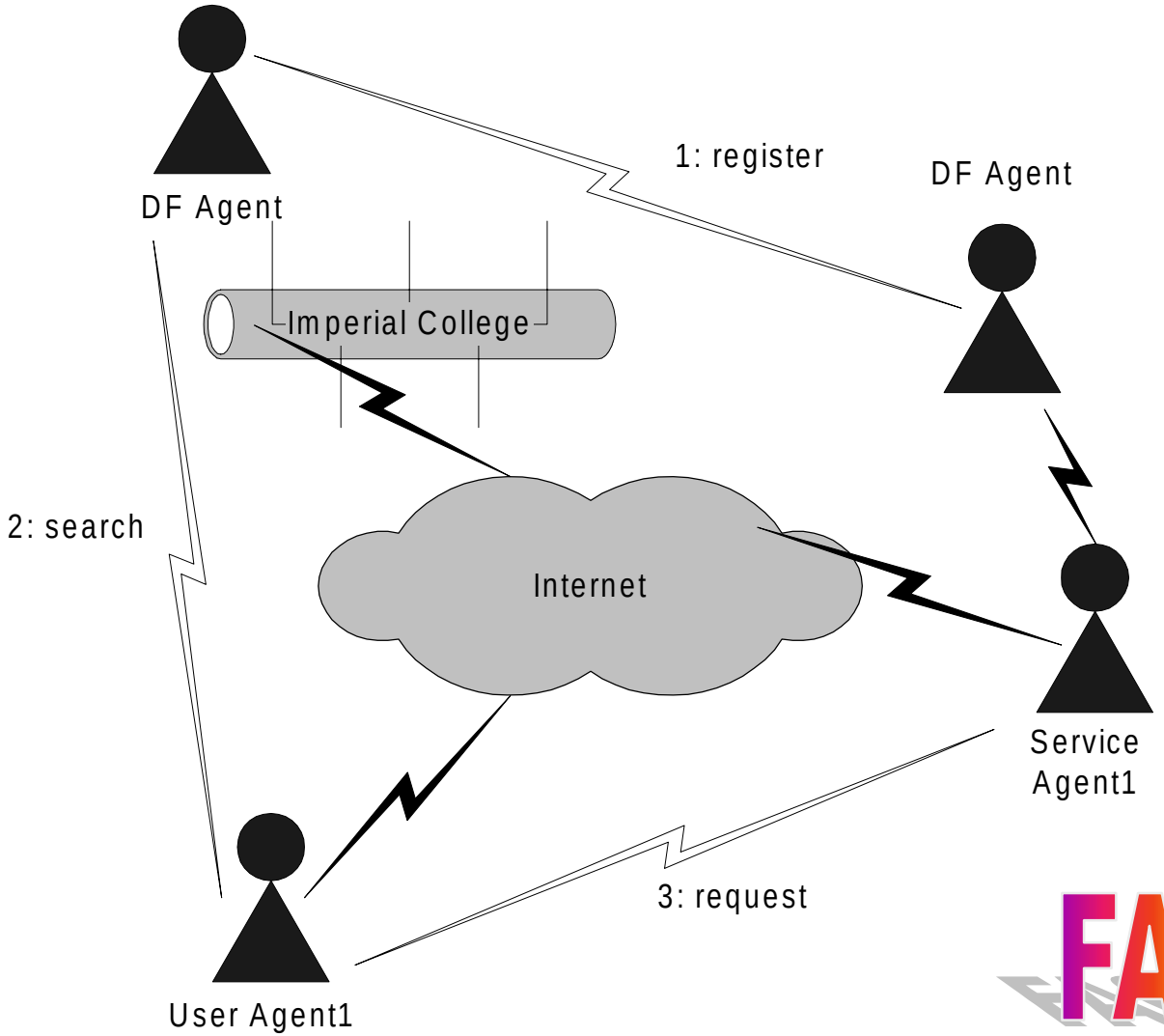
FIPA-OS - Requirements

- **Java 1.2**
- **Third-party software**
  - XML Parser, e.g. IBM XML4J
  - RDF Parser/Validator, e.g. W3C SiRPAC
  - Web server e.g. Apache (for IOR distribution only)
- **Recommended minimum specification PC**
  - Pentium 166 processor
  - 64 MB memory
  - 4 MB disk space

Installation Process

- **Automatic installation and platform configuration**
  - Works straight ‘out of the box’
  - Platform configured to run locally
  - Automatically locates required third party software
  - Installs on Windows or UNIX
- **Refresh or upgrade installation**
  - Non-destructive
  - Existing settings migrated to new installation
  - Your messages, code and classes are backed up

FIPA-Net: FIPA-OS Live Internet Service



FACTS

FIPA-OS Status

- **v1.01a released September 1999**
- **v1.02 released October 1999**
- **v1.03 released January 2000**
- **v1.04 planned for March 2000**

- **Growing global user base from Academia and Industry**

Example FIPA-OS Users

- **European Collaborative projects:**
  - FACTS
  - Cameleon
  - MAPPA
  - Shuffle
- **Universities**
  - Imperial College
  - QMW
  - EPFL
  - Nottingham Trent

Further Information

FIPA-OS is a Open Source implementation of FIPA and is available for free.

- **FIPA-OS**
  - http://www.nortelnetworks.com/fipa-os
  - **Email user group**
    - http://fipaos.listbot.com
    - fipaos@listbot.com

Backup

Conversation IDs

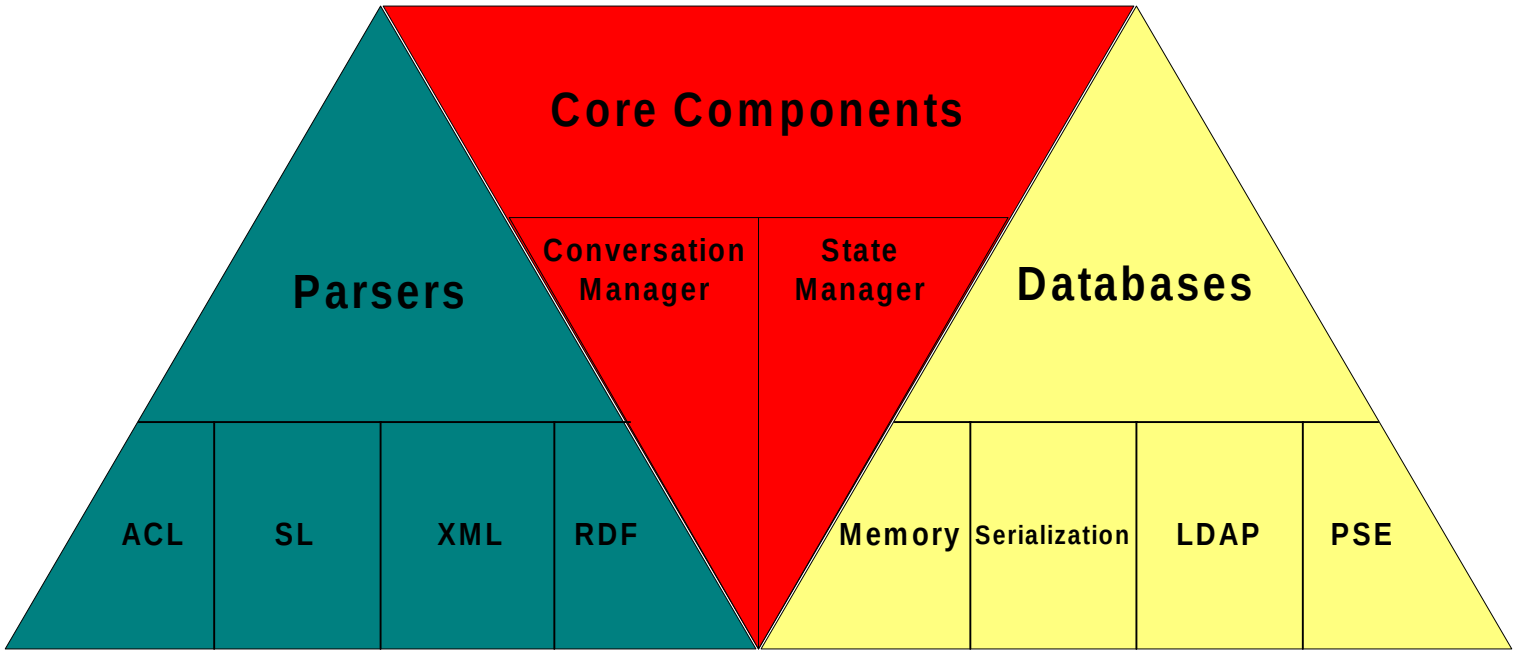
- **FIPA-OS adds semantics to Conversation IDs**
  - AgentGUID + current time in milliseconds + count
  - AgentGUID ensures an agent is unique on a platform instance
  - Current time makes sure that the ID is unique if two platforms on same machine have an agent of same name
  - The count ensures no two conversations can have same ID being created at the same time
- **Conversation IDs use in Conversation Management**
  - Conversation IDs uniquely identify a single conversation between agents, ‘reply-with’ and ‘in-reply-to’ define the message sequence for a single conversation
- **Interoperability with non-FIPA-OS platforms**
  - Conversation Ids received from remote platforms are assumed to be unique

Conversation Manager

- **Provides per conversation handling of agent interactions**
  - Conversation state is maintained automatically on behalf of the agent
  - Provides a “memory” of previous conversations
  - Automatic responses for malformed messages
- **Ensures correctness of agent conversations following FIPA defined protocols**
  - Default responses to incoming messages not following protocol correctly
  - Outgoing messages prevented from being sent if conversation protocol isn’t being followed

FACTS

Agent “Shell” - Overview



FACTS

FIPA Compliance Points

- **FIPA98 Agent Management**
- **FIPA97 v2 Agent Communication**
- **FIPA97 v1 Agent/Legacy Integration**
- **FIPA99 Agent Management**
  - (in part, aiming to replace support for FIPA98 Agent Management)

FIPA Platform Interoperability

- **Alcatel's 3AP**
- **GMD's Grasshopper extension**
- **CSELT's JADE**
  - Note: unreleased version used that supports FIPA98 Agent Management
- **Broadcom's ASL**

FACTS

Developing Agents

- **FIPA-OS Agent API**
  - Messaging methods to send and receive messages
  - Platform registration methods for local and remote platform agents
    - Optional dynamic callbacks for ‘live’ registration information
  - Automatic conversation management
    - Ensures correct protocol usage
  - State management
- **Generic Agent**
  - ‘Empty’ agent that just registers with the platform
  - Base new agents on this code – good for learning API
- **IO Test Agent**
  - Send test messages to agents and analyse the responses

FACTS

Changing FIPA-OS (1/2)

- **Compiling classes to replace classes in the distribution**
  - Automatically re-compile whole the whole of FIPA-OS and replace all pre-compiled classes
  - Manually re-compile individual classes to replace those distributed with FIPA-OS
- **Build process**
  - Automated build process to simplify FIPA-OS development

FACTS

Changing FIPA-OS (2/2)

- **FIPA-OS Community**
  - Email discussion group for discussing issues and ideas for future expansion
    - Email: fipaos@listbot.com
    - Email Archive: http://fipaos.listbot.com
  - In the spirit of Open Source bug fixes and feature improvements should be delivered to the FIPA-OS Community
    - Email code to: fipaos@listbot.com
  - Nortel Networks manage the integration of new code and makes regular controlled code releases