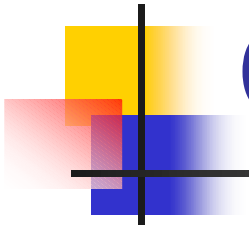
MOVE:
Component Groupware Foundations for Collaborative Virtual Environments

Pedro García, Antonio Gómez Skarmeta, Oriol Montalà, Carles Pairot, Robert Rallo
pgarcia@etse.urv.es, skarmeta@fcu.um.es, {omp.ei, cpg.ei}@estudiants.urv.es, rrallo@etse.urv.es

Department of Computer Science, Universitat Rovira i Virgili
Department of Computer Engineering, Universidad de Murcia



Outline

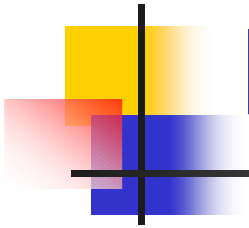
- Motivation
- Related work
- System Architecture
- The ANTS CSCW Framework
- MOVE Architecture
  - Session and Zone Management
  - Shared Artefacts and State Propagation
  - Coordination and Consistency
  - Awareness
- MOVE Performance Issues
- Conclusion and Future Work

Motivation

- **Current situation**
  - Advances in networking technologies and protocols, DBs, Computer Graphics and Display Technologies
  - Consequence: Many 3D CVEs have emerged
- **What is MOVE ?**
  - 3D CVE. Interaction with other users or with shared artefacts
  - Tested in the Catalonian Internet2 project
  - Deemed for educational purposes
  - Developed using open technologies: VRML, H-Anim, Java, ...

Motivation

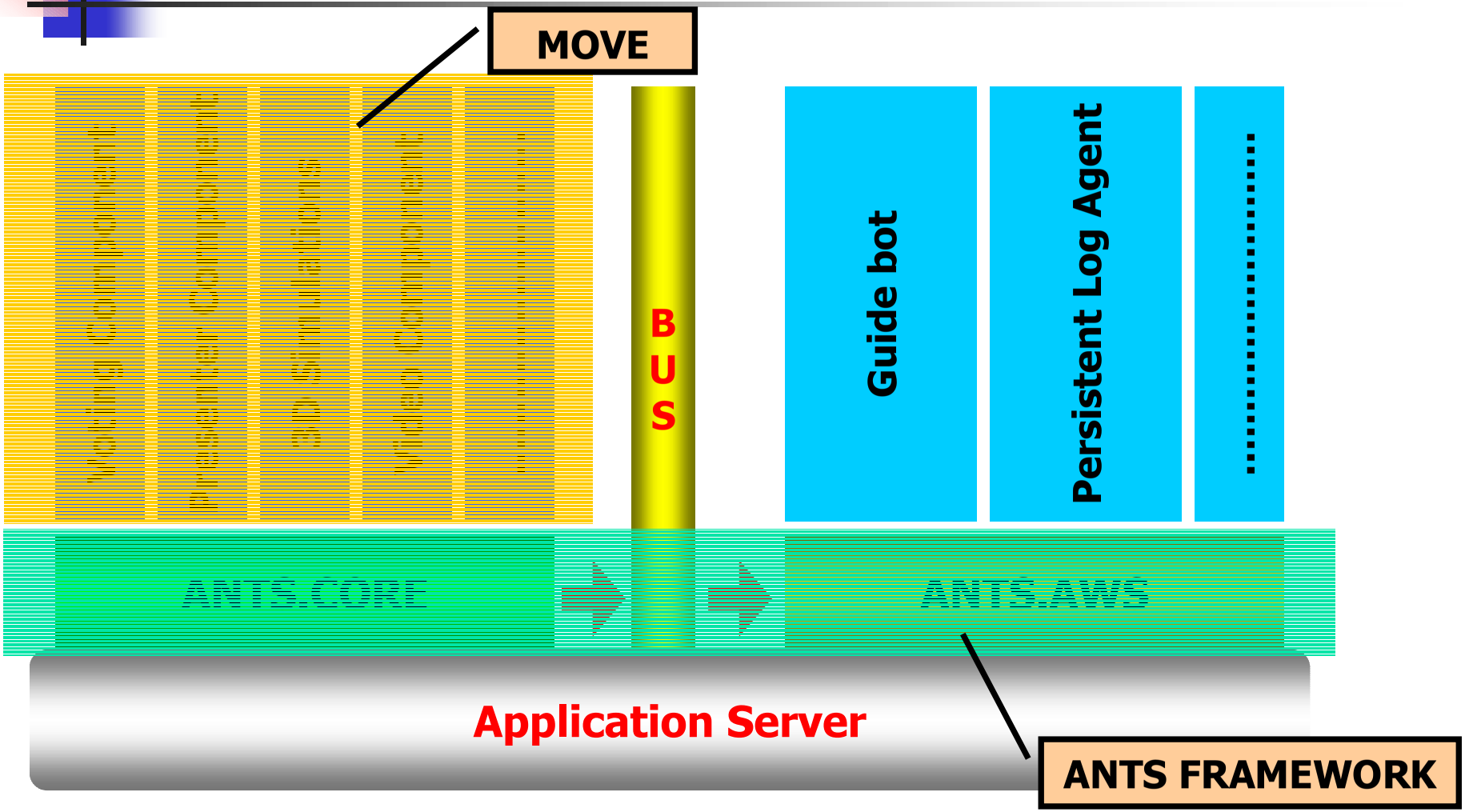
- **Problems in existing 3D CVEs:**
  - High complexity in development. Should be scalable and support many concurrent users. Addition of new components: **Component Reusability**
  - **Data Extraction**: Were not thought for retrieving data in an easy way
- **MOVE** uses the **ANTS CSCW Framework:**
  - Facilitates development of collaborative components
  - Smooth transition from local to distributed apps



Motivation

- **How have we solved the previous problems?**
  - Definition of a **Component Model** based on the standard *JavaBeans* specification
    - Hides complexity to users, providing
      - Transparently Remote Persistence
      - Distributed Events
      - XML Component Descriptors & Packaging
  - Definition of a dedicated **Awareness** (AWS) and **Event Monitoring** service
    - Also, an agent system that reacts to events triggered in the bus has been implemented
    - Any kind of information can be obtained!

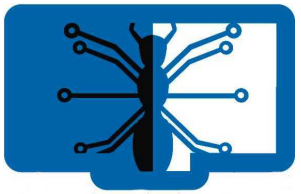
System Architecture



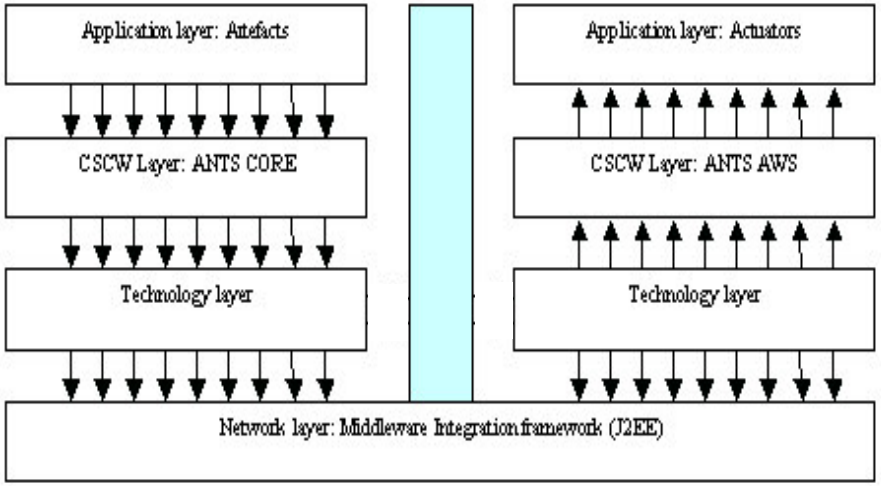
Related Work

- **Component Architecture Approaches:**
  - NPSNET-V: Components loaded at runtime
  - NPSNET Bamboo: Code modules operate in X-platform and X-language manner
- **Awareness Services Approaches:**
  - TOWER: Event analysis visualized with Blaxxun
  - MASSIVE-3: Record & Replay mechanisms
- **Declarative Component Approaches:**
  - CONTIGRA and Jamal: Both define XML-based mark-up languages for 3D component description & assembly

The ANTS CSCW Framework

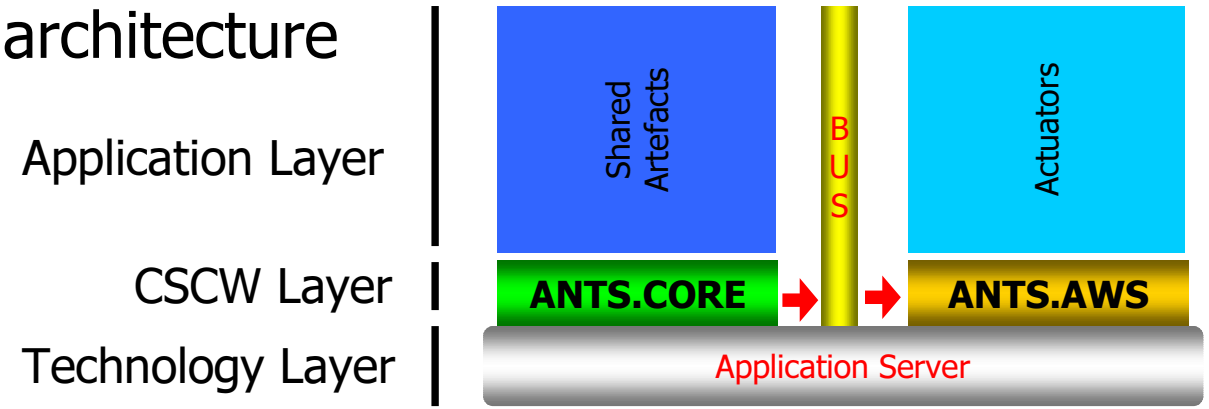


- Provides a Generic Multi-User Collaborative Framework
- Key piece: The Collaboration Bus



The ANTS CSCW Framework

- The collaboration bus
  - Is not only an event dispatcher
  - Is a higher level abstraction constructed on top of publish/subscribe middleware.
    - Creates a state propagation system for shared components transparent to developers.
    - Also creates a mediator system. 3rd-party actuators can react to environment events
- 3-layered architecture

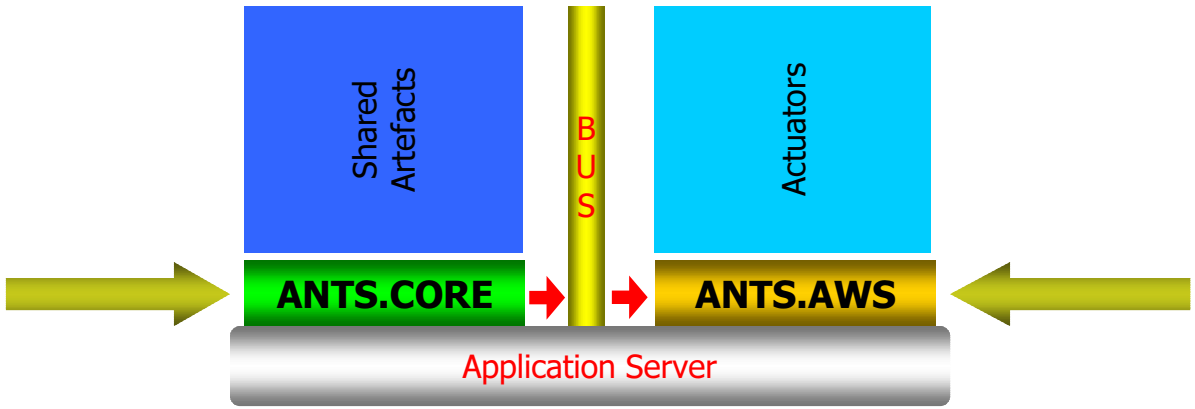


The ANTS CSCW Framework

- **Application layer** (2 extension hooks)
  - Provides development of new collaborative components using the *JavaBean* specification
  - Provides awareness actuators that react to information events produced in the Framework
  - We have created a **Component Model** comprising
    - Persistence & Events
    - Coordination
    - Customization, Introspection & Packaging

The ANTS CSCW Framework

- **CSCW Layer**
  - Set of Collaborative Services offered to the Application Layer.
- Modules
  - Container Runtime Module (**ANTS.CORE**)
  - Awareness Services Module (**ANTS.AWS**)

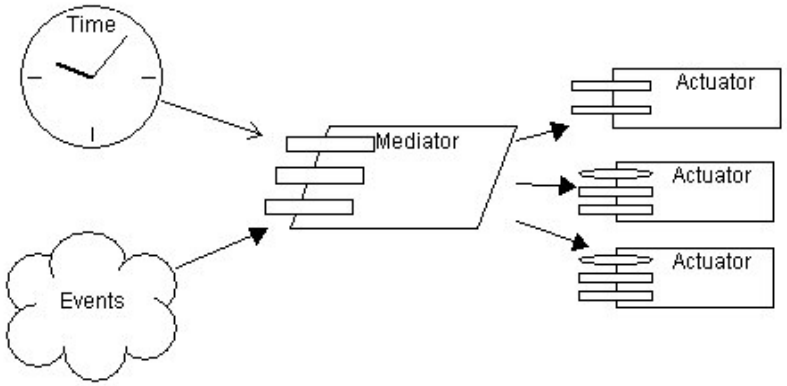


The ANTS CSCW Framework

- **ANTS.CORE**
  - Support for **Sessions, Shared Artefacts, Coordination Control, Security** and a **Seamless Security Model**
  - MOO Influence
    - Session = *Place*
    - Artefacts = *Things*
  - **Place**
    - Represents the shared session. Contains users, components and links
    - Provides several methods: send/subscribe events, get connected users, get links, ...
    - Dynamically loading of *JavaBean* components to the shared context.

The ANTS CSCW Framework

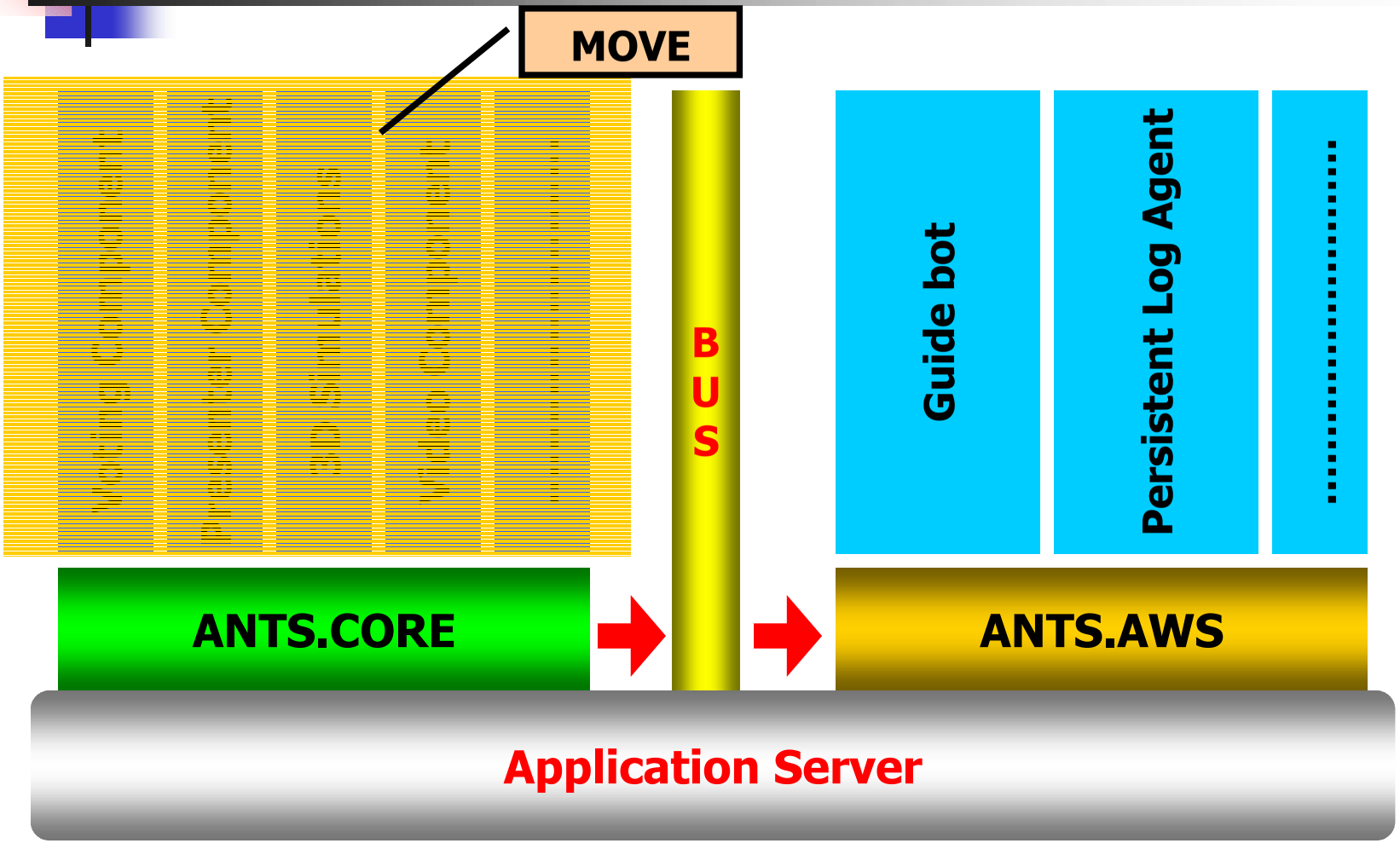
- **ANTS.AWS**
  - Enables triggering of a **set of actuators** in response to **events** produced in the system
  - Integrated with the CM through Component's XML descriptors
  - AWS follows a **Mediator design pattern**
    - Sensors, Mediator and Actuators



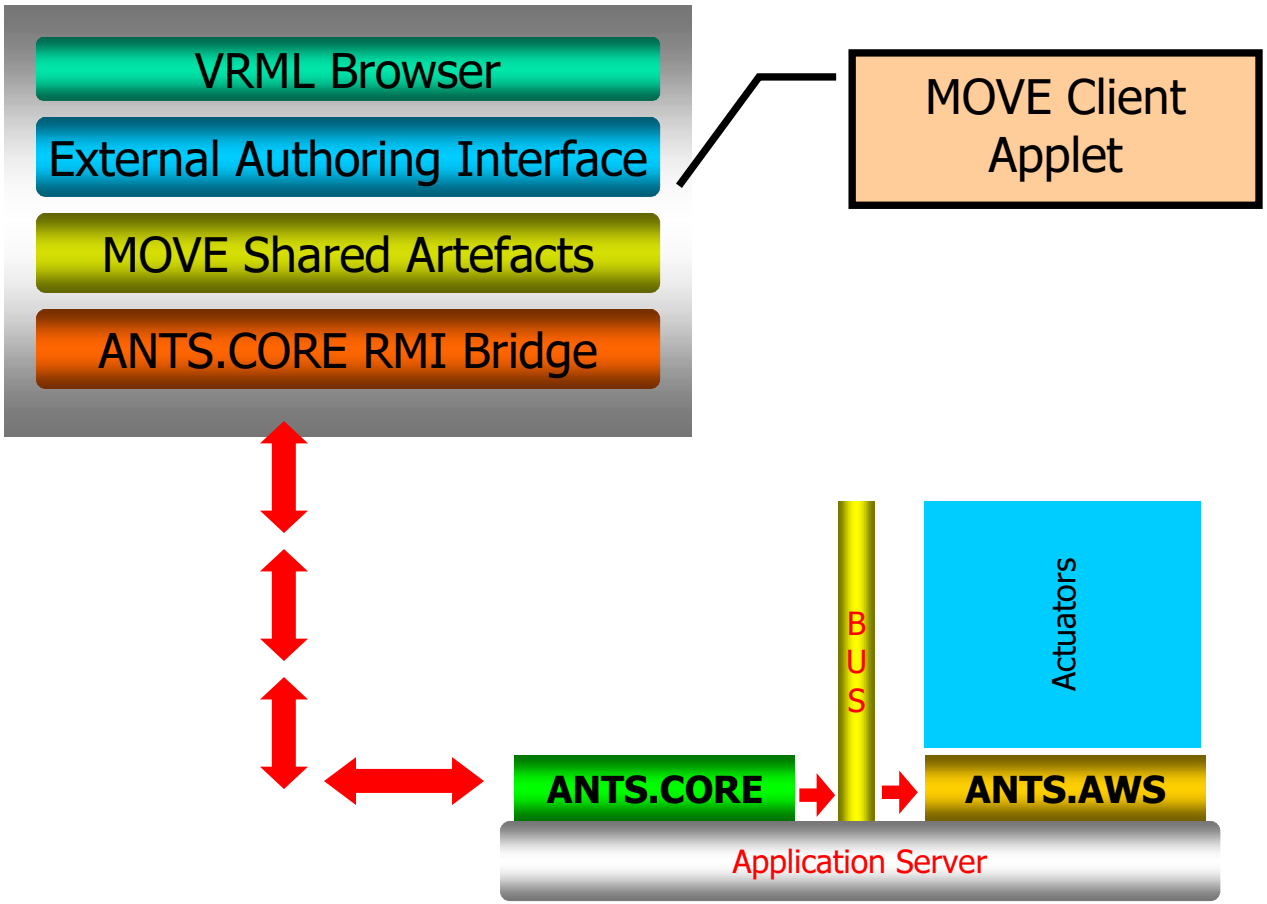
The ANTS CSCW Framework

- **Technology Layer**
  - Solid infrastructure providing security, scalability, transactions and performance needed
  - **J2EE** Platform was chosen as technology infrastructure
    - Infrastructure and System independent code
    - Lets us base on open specifications and components
    - Vendor independent
  - Concerning MOM
    - **DSTC's Elvin Notification Service**
    - Façade API that lets us choose between **JMS** and **Elvin**

MOVE Architecture



MOVE Architecture

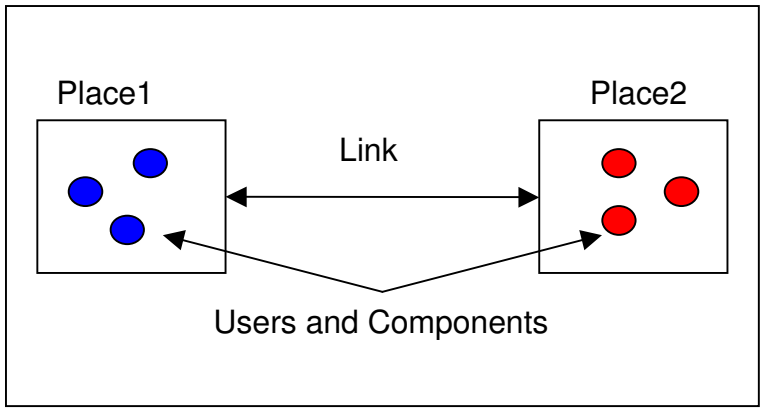


MOVE Architecture

- **Essential Collaborative Services**
  - Session and Zone Management
  - Shared Artefacts and State Propagation
  - Coordination and Consistency
  - Awareness

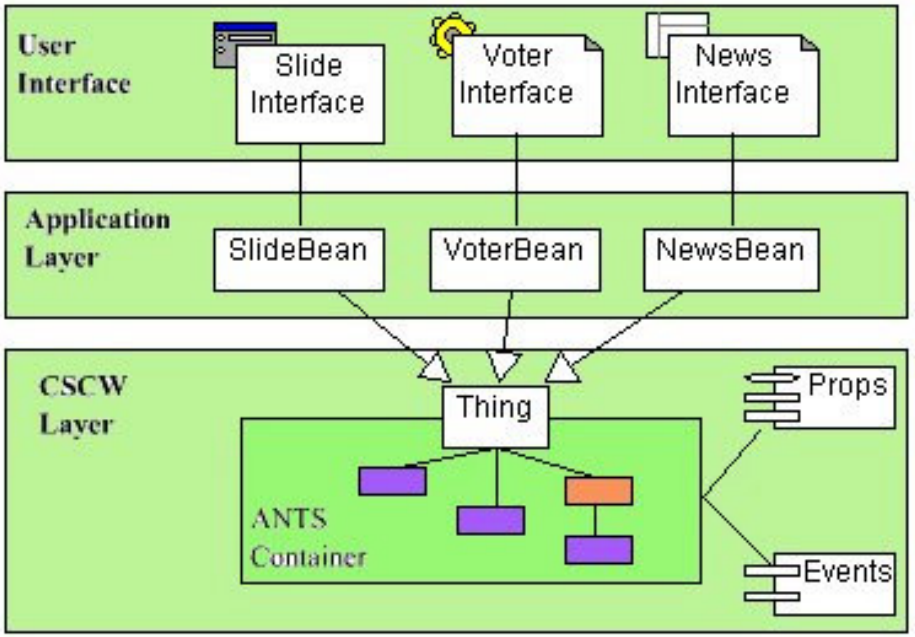
MOVE Architecture

- **Session and Zone Management**
  - Use of the *Place* concept to determine the environment structure
  - Session is identified with *place*
  - *Places* are similar to *Locales* in SPLINE
  - Portals are used for linking places



MOVE Architecture

- **Shared Artefacts and State Propagation**
  - MOVE is based on a modular architecture for every component, following the MVC paradigm

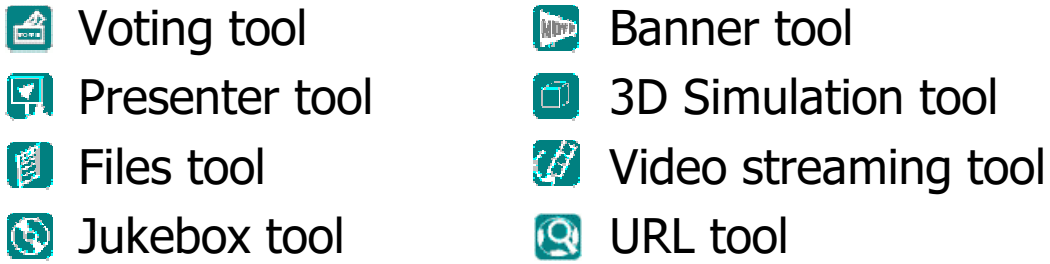


MOVE Architecture

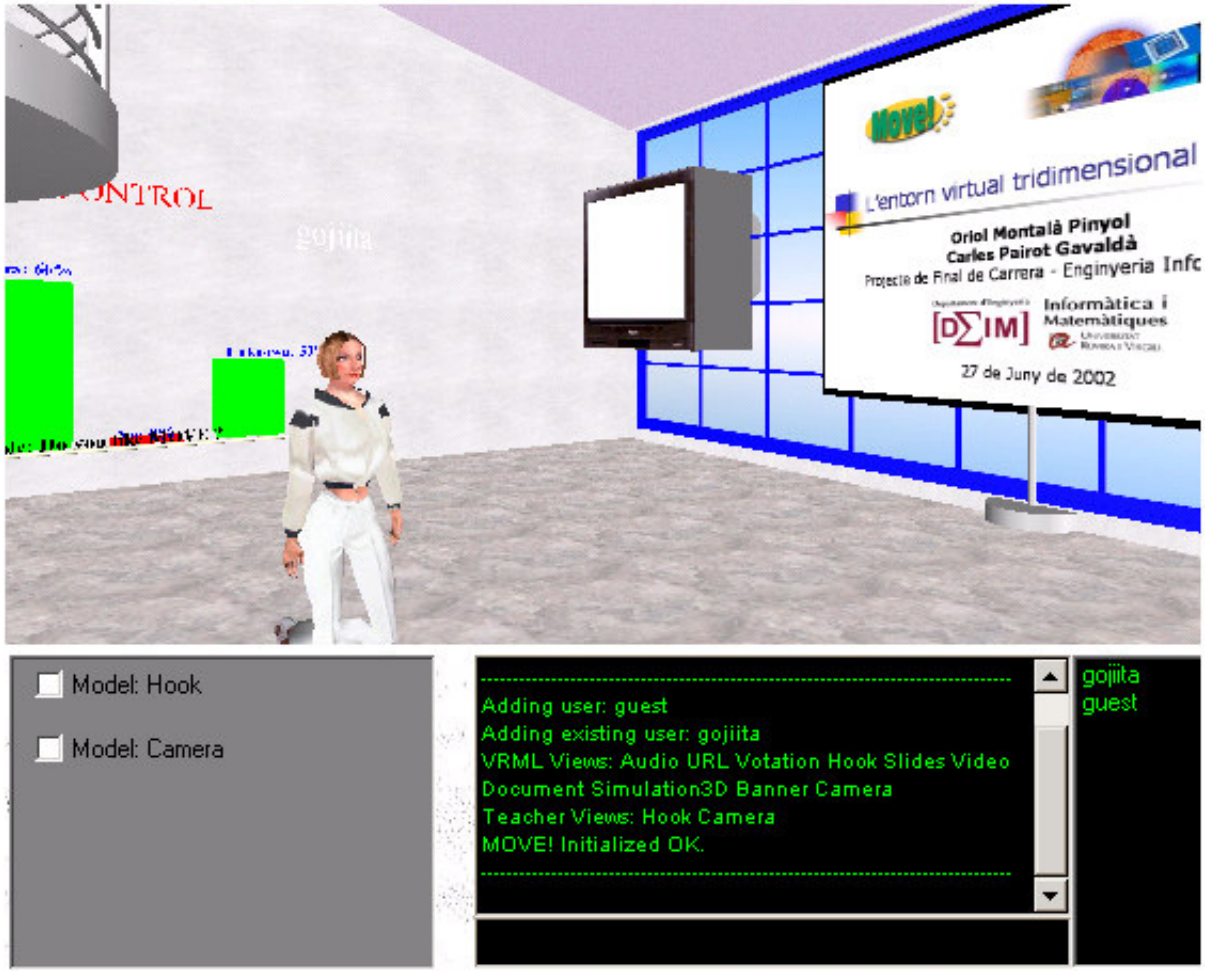
- **Shared Artefacts and State Propagation (continued)**
  - **Main aim**: design a system so as it is easily extensible and reusable
  - New components contain an XML descriptor with all required information
  - New components are JAR-packaged and deployed on the ANTS Server

MOVE Architecture

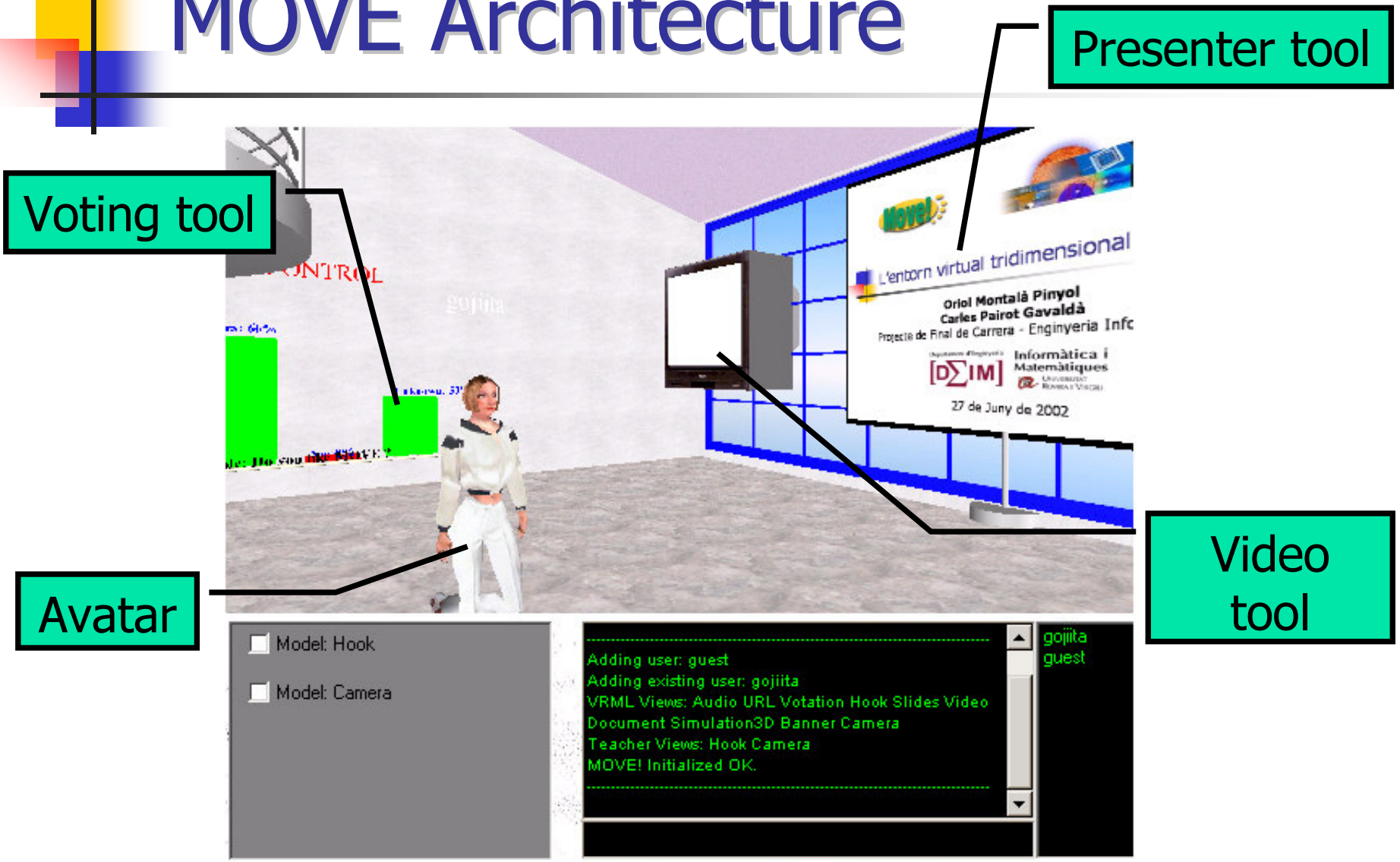
- **Shared Artefacts and State Propagation (continued)**
  - We have not taken care of state propagation, nor persistence: **ANTS CSCW does it for us!**
  - We have developed a set of artefacts:



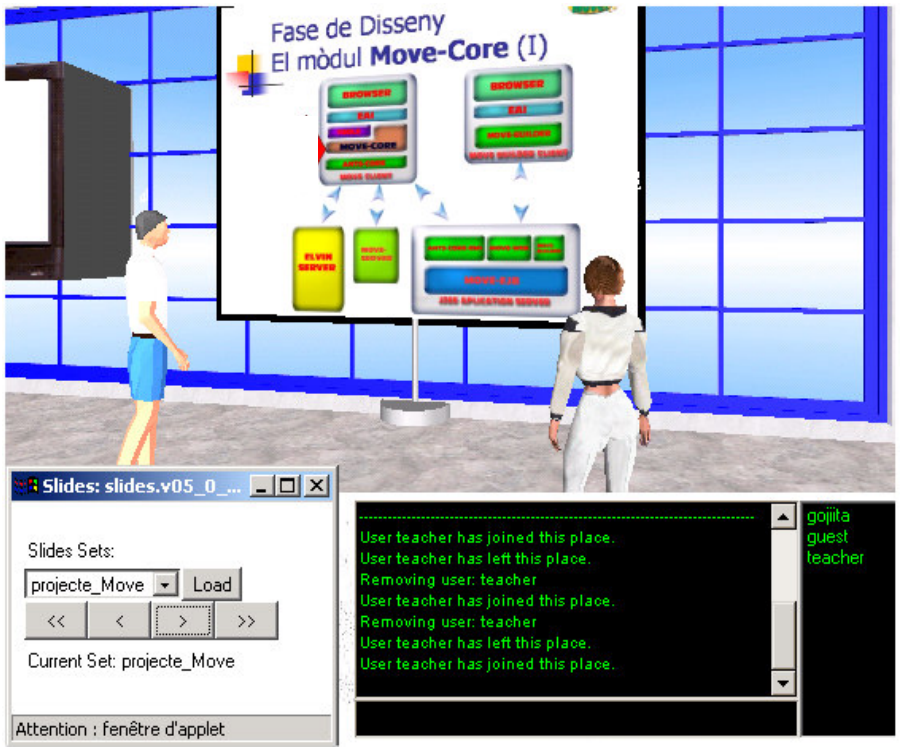
MOVE Architecture

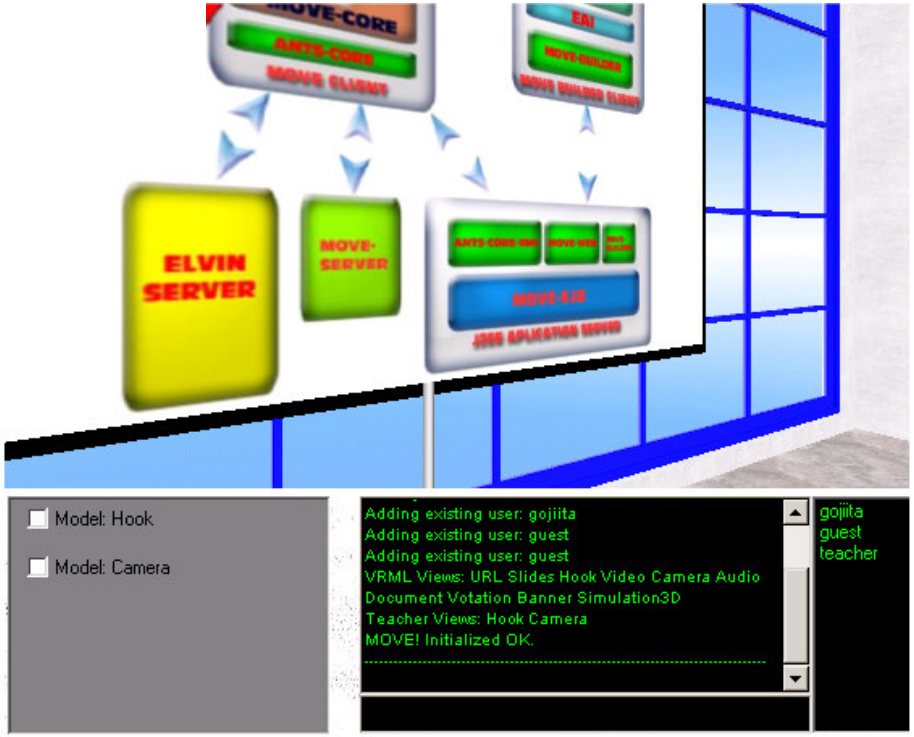


MOVE Architecture



MOVE Architecture

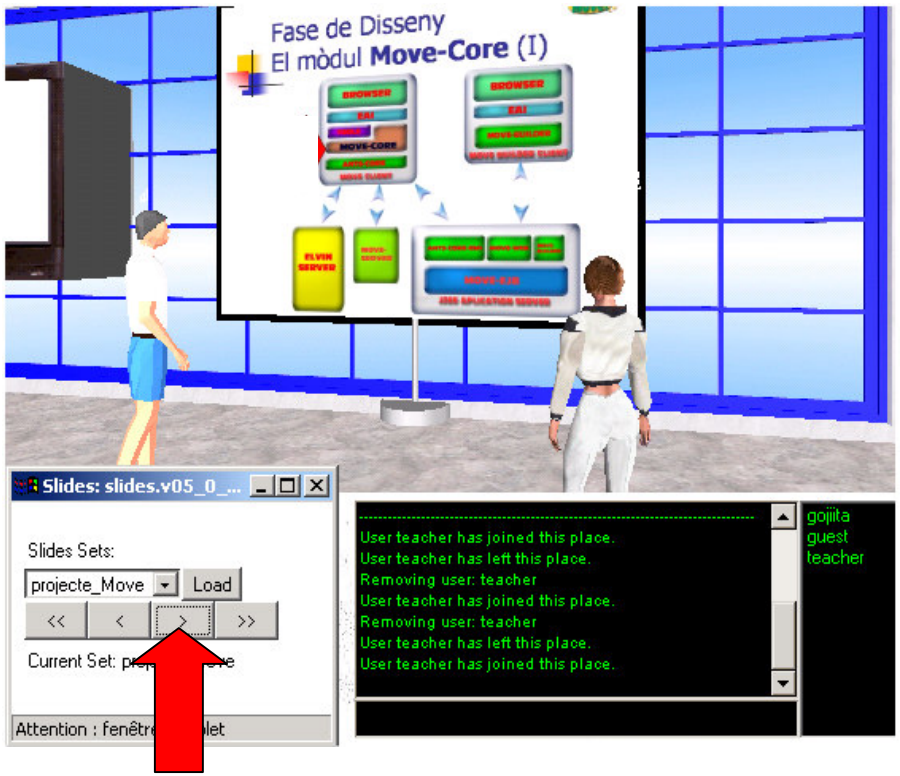




User 1

User 2

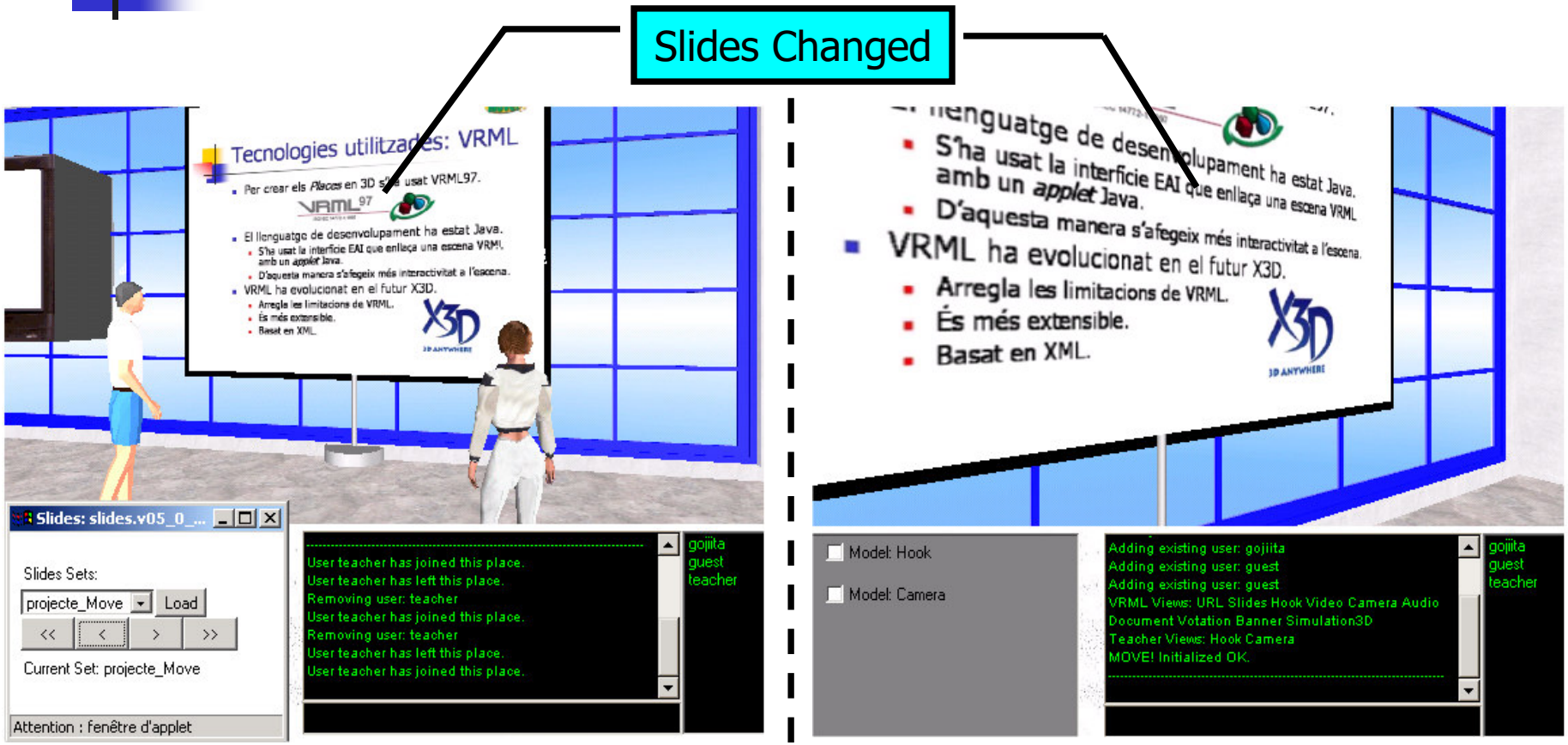
MOVE Architecture



User 1

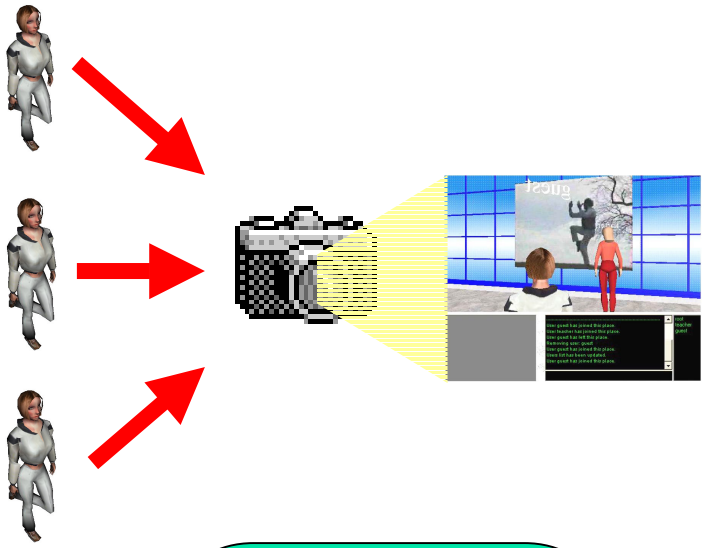
User 2

MOVE Architecture

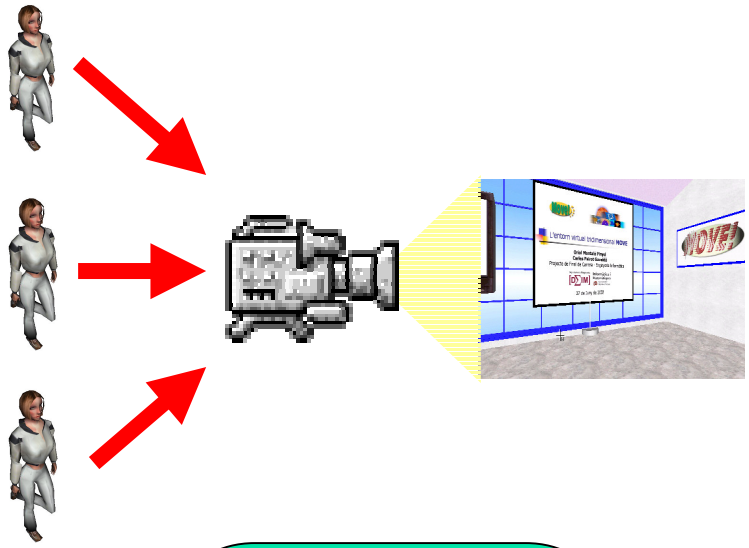


MOVE Architecture

- **Shared Artefacts and State Propagation (continued)**
  - Educational Environment
  - User's attention grabbing tools designed



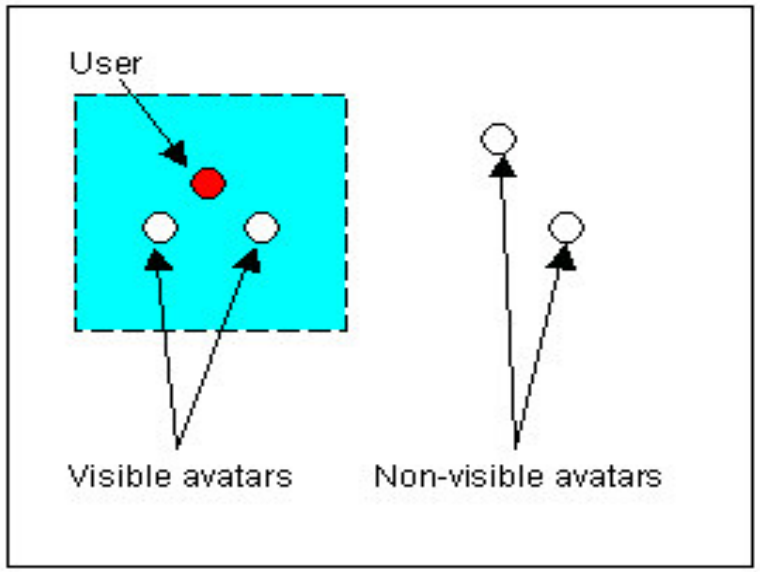
Camera tool



Hook tool

MOVE Architecture

- **Shared Artefacts and State Propagation (continued)**
  - However, not all components can use state propagation mechanisms of ANTS CSCW
  - Example: **Avatar Movement Component**

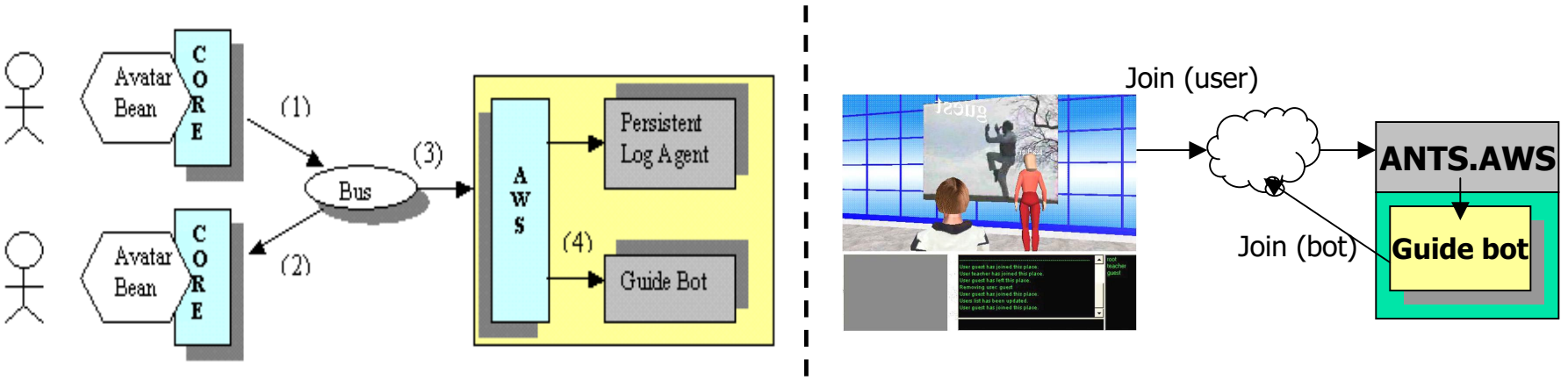


MOVE Architecture

- **Coordination and Consistency**
  - **Coordination policies** must be supported by software
  - Can be categorized in roles for access control and concurrency control
  - MOVE's authorization is based on a role model
  - Future work: Concurrency control mechanism. **Token-based lock component** → Transport artefacts from 1 place to another

MOVE Architecture

- **Awareness**
  - An awareness platform should provide data acquisition from the running environment
  - MOVE uses ANTS CSCW awareness and monitoring services (ANTS.AWS)
  - Example: **guide bot**



MOVE Performance Issues

- After performing massive tests:
  - Critical point: **Client** and not the ANTS platform or server-side middleware
  - Implementation of distance-based algorithm. Event discarding
  - Future work: implementation of dead-reckoning algorithms
  - Even though, MOVE scales up smoothly with 200 users in a shared session

Conclusion & Future Work

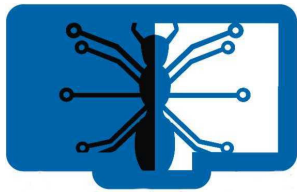
- Major Features
  - **MOVE**: CVE constructed on top of a Component Groupware Framework
  - Provide consistent Groupware Foundations to the development of VEs
  - Provide **Extensibility** to all framework levels

Conclusion & Future Work

- Future Work
  - Improve coordination & consistency control mechanisms
  - Improve performance
  - Apply data analysis techniques to better understand collaboration flows
  - Extension of the agent system
    - Prototype developed using *Python*
    - A Capture & Replay agent has been developed

Thank you for your attention!

- If you wish to contact us:
  - Pedro García
    pgarcia@etse.urv.es
  - Antonio Gómez Skarmeta
    skarmeta@fcu.um.es
  - Oriol Montalà
    omp.ei@estudiants.urv.es
  - Carles Pairot
    cpg.ei@estudiants.urv.es
  - Robert Rallo
    rrallo@etse.urv.es



- MOVE Website:
  http://ants.etse.urv.es/move