

A Multi-Agent Platform for Automatic Assignment Management

Abelardo Pardo



The Gradient Group
Carlos III University of Madrid (UC3M)
Department of Telematic Engineering
Madrid, SPAIN
abel@it.uc3m.es

Outline

- **Introduction**
- Related Research
- Overview of the Proposed Architecture
- The Student, Professor and Grader view
- Sample Project
- Conclusions and Future Work

Computer Aided Assessment

- Computer science courses have a **significant** amount of projects.
- Classes tend to have **large** number of students.
- The need for **Computer Aided Assessment** is becoming crucial.
- **Automatic Assessment** **enhances** the overall education experience.

Information and Communication Technology

- ICT has significantly changed the **context** for Automatic Assessment tools.
- Distance Learning classes require **remote** assessment.
- **Conventional classes** may also benefit from remote assessment.
- Automation is present in most of the **administrative tasks** in educational institutions.

Problems faced by Automatic Assessment

- Several commercial products already support **some form** of automatic assessment.
 - **Multiple choice.**
 - **Fill in the blank.**
 - **Randomization of answers.**
 - **Statistical analysis of results.**
- These tools provide fully automatic grading of a **very specific** type of assignment.
- These assignments represent a **subset** of the evaluation requirements in a computer science course.
- Assignments require certain degree of **human intervention.**

The Proposed Platform

- The proposed platform aims at satisfying the following goals:
 1. **Versatility**: Support a large variety of assignment types and grading policies under the same paradigm.
 2. **Scalability**: Provide a **scalable** framework to cope with large classes of students possibly located in different geographical regions.
 3. **Integration**: Provide **full connectivity** with other administrative tools already present in educational institutions.

Outline

- Introduction
- **Related Research**
- Overview of the Proposed Architecture
- The Student, Professor and Grader view
- Sample Project
- Conclusions and Future Work

Related Research

- Tools for automatic assessment started as a **collection** of scripts applied to the submissions.
- Early tools focused mainly in the **administrative aspects** of the assignment.
- **Processing** of an assignment was independent of its **content**.
- **State of the art tools** provide functionality for designing quizzes, manage question databases and integrating with the rest of the material included in a course.
 - CourseMaster (formerly known as Ceilidh)
 - TRAKLA
 - The Castle Toolkit
 - WebCT
 - WebMCQ
 - QuestionMark
- Other tools such for example **Scheme-Robo** are specialized in programming assessments written in a specific programming language.

Semi-automatic Platforms

- English et al. noted the need for a platform to integrate **several assessment tools** into a common framework.
- Jackson outlined a semi-automatic approach that executes **automatic tasks** whenever possible and then prompts the tutor for **more advanced judgments**.
- Shi et al. presented a **multi-agent platform** for distance learning although no concrete assessment application was specified.

Evaluation Criteria for Automatic Assessment Systems

- **J. Dalziel enumerated five criteria to evaluate automatic assessment systems:**
 - 1. Ease of use from different views (student, professor, grader).**
 - 2. Required level of technical proficiency.**
 - 3. Hardware/Software system requirements.**
 - 4. Data collection capabilities.**
 - 5. Platform cost.**
- **Additional aspects considered in the presented platform:**
 - Scalability**
 - Versatility**
 - Integration**

User Profiles

- The tool considers three user profiles

1. **Student:** Receives the assignment and submits answers.

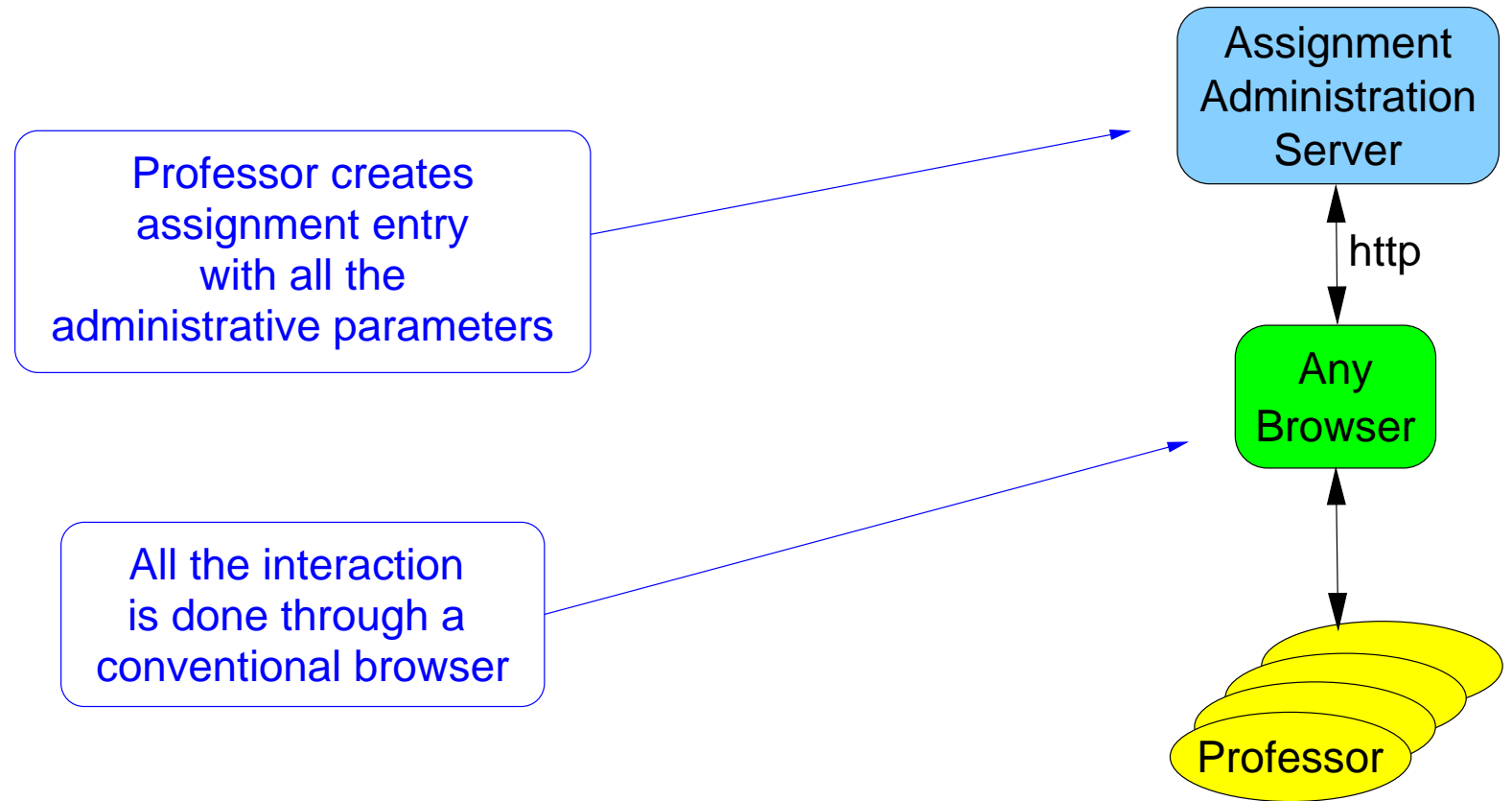
2. **Professor:** Designs the assignment and controls the administrative aspects.

3. **Grader:** Evaluates the submissions.

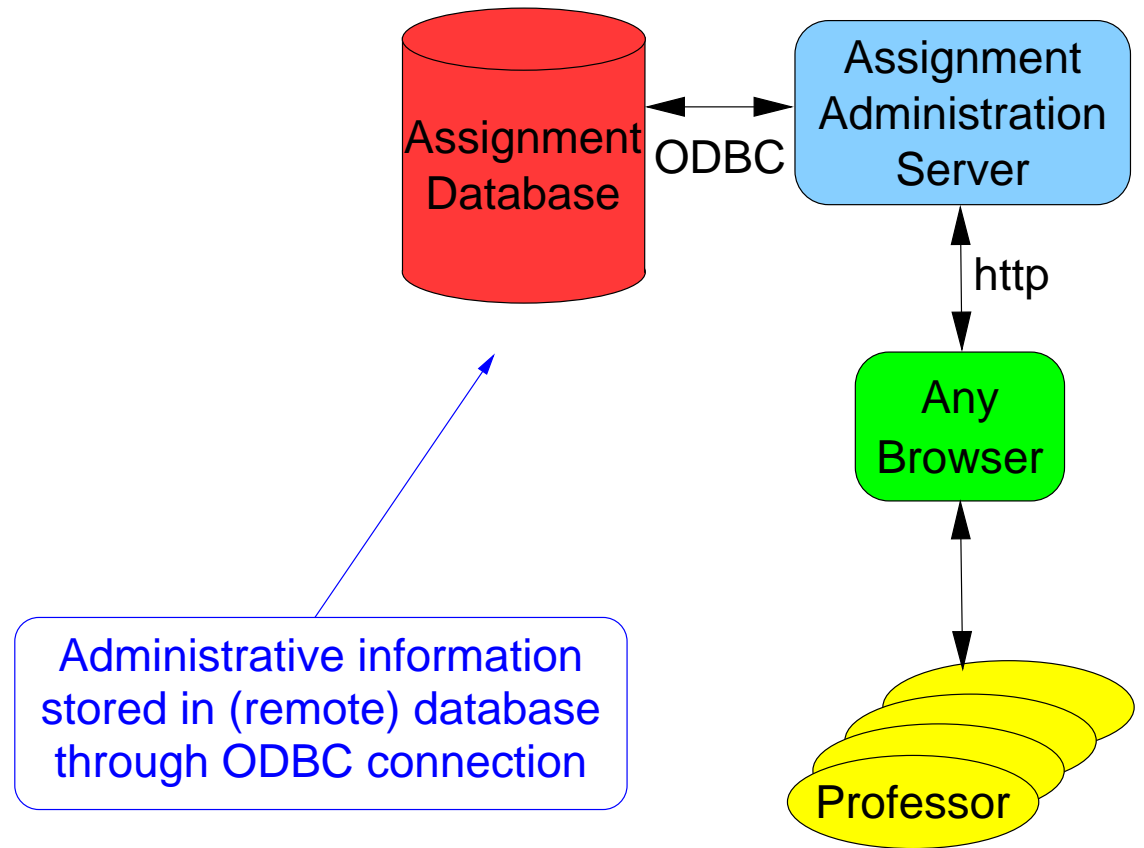
Outline

- Introduction
- Related Research
- Overview of the Proposed Architecture
- The Student, Professor and Grader view
- Sample Project
- Conclusions and Future Work

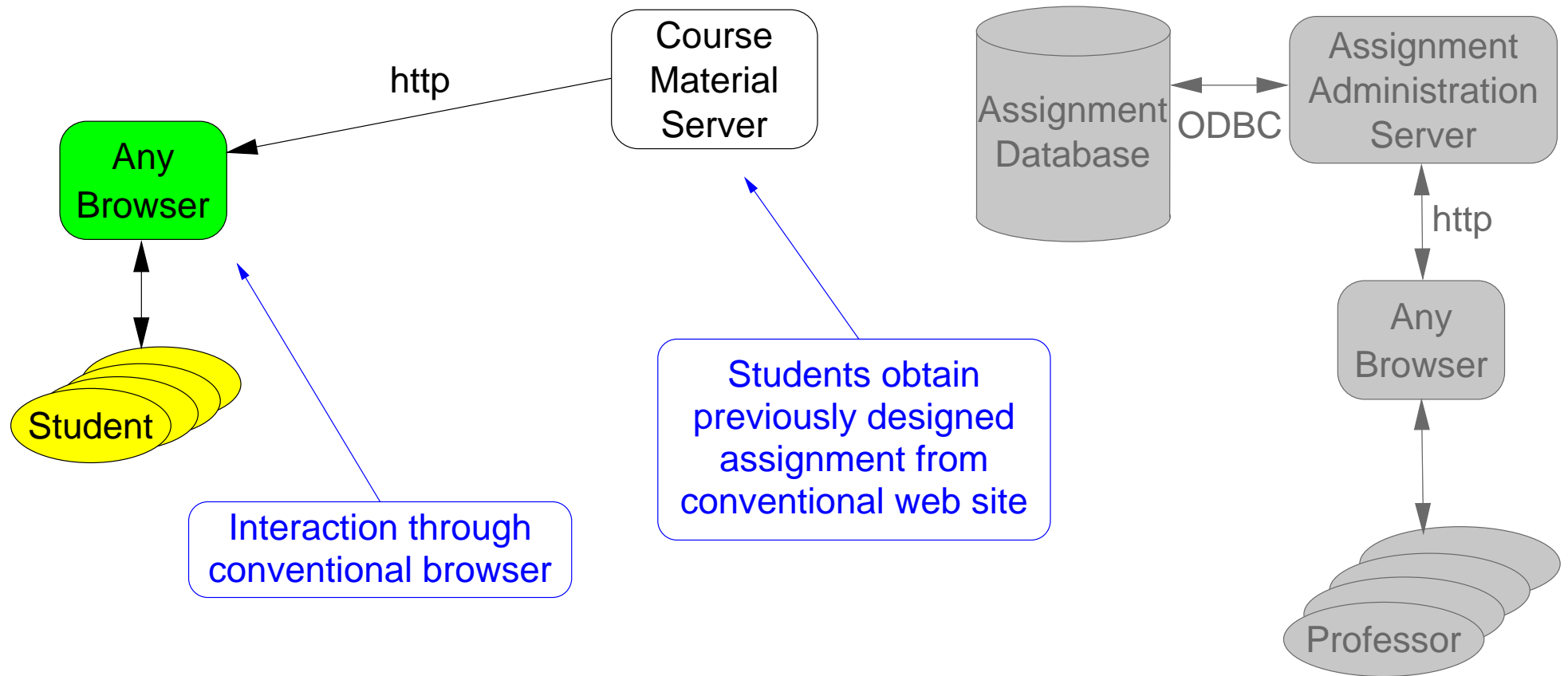
Platform Architecture



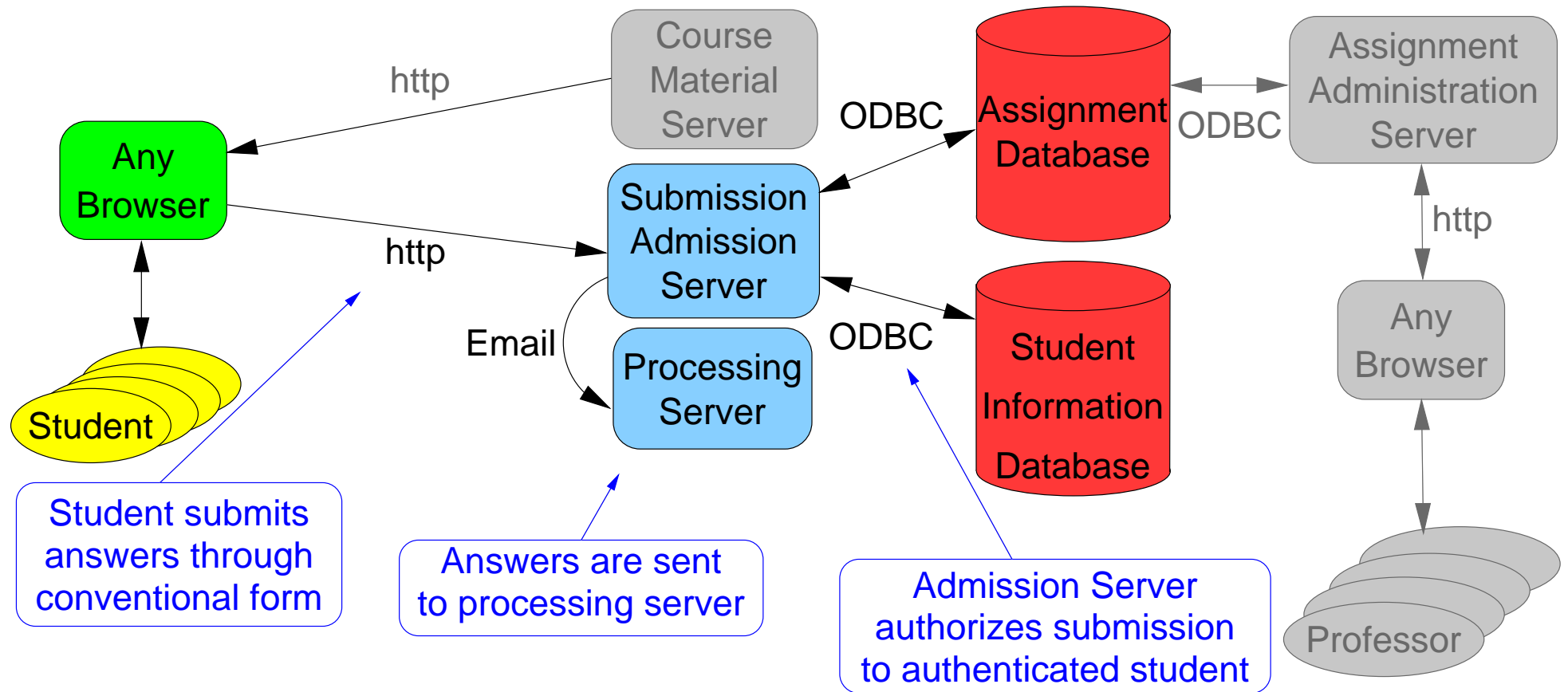
Platform Architecture



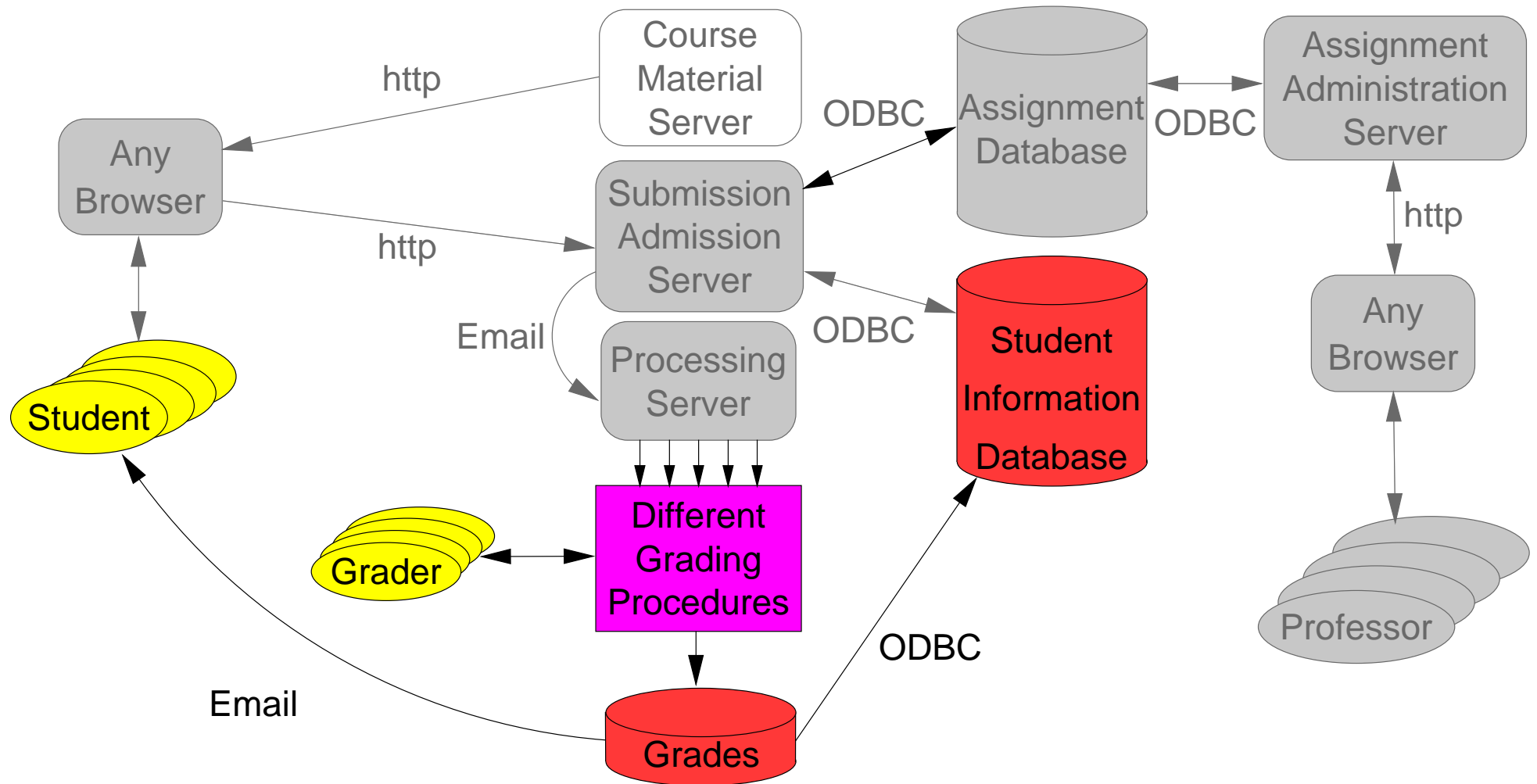
Platform Architecture



Platform Architecture



Platform Architecture



Grading Procedures

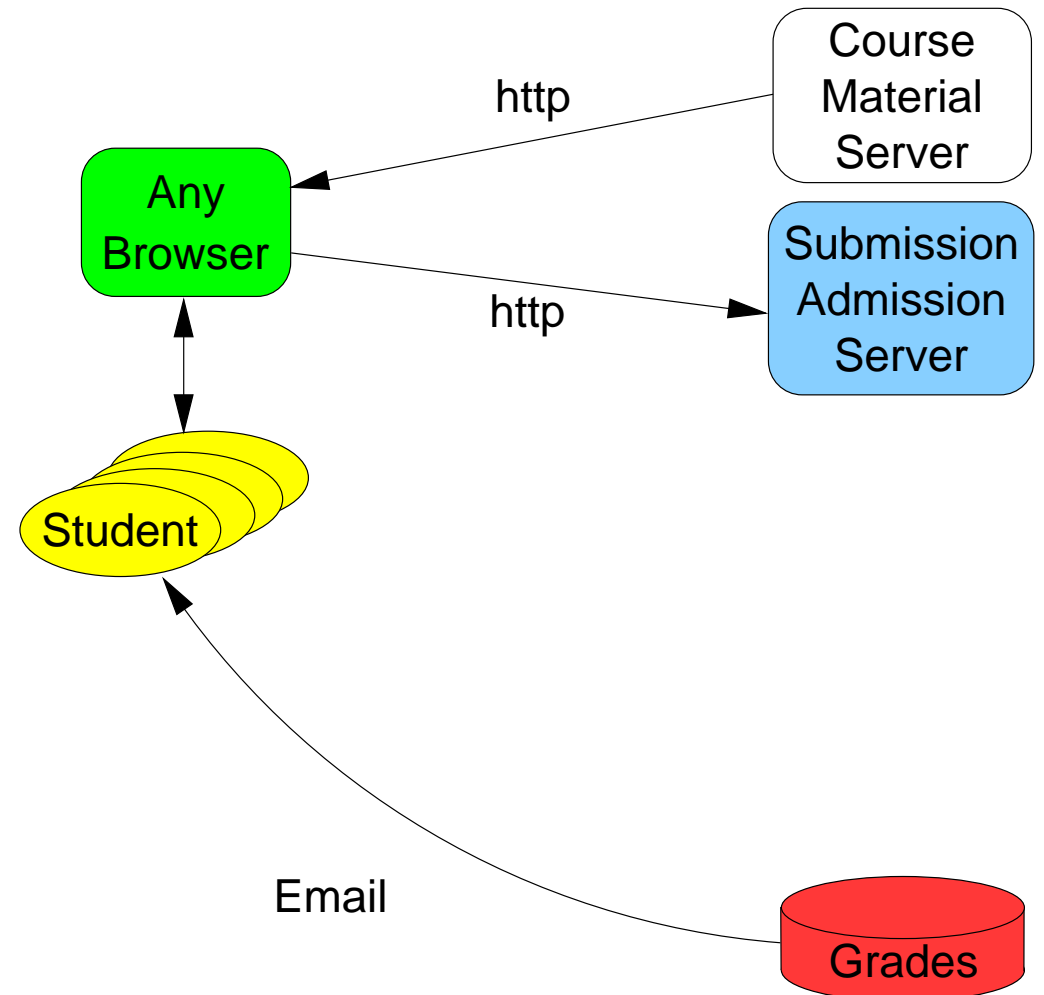
- Submission data is stored in **XML**.
- Data is made available to **grading procedures**.
- Graders may edit the submission with **conventional editors**.
- Automatic procedures may access data through **XML manipulation**.
- Basic administrative procedures are **already implemented**.

Outline

- Introduction
- Related Research
- Overview of the Proposed Architecture
- **The Student, Professor and Grader view**
- Sample Project
- Conclusions and Future Work

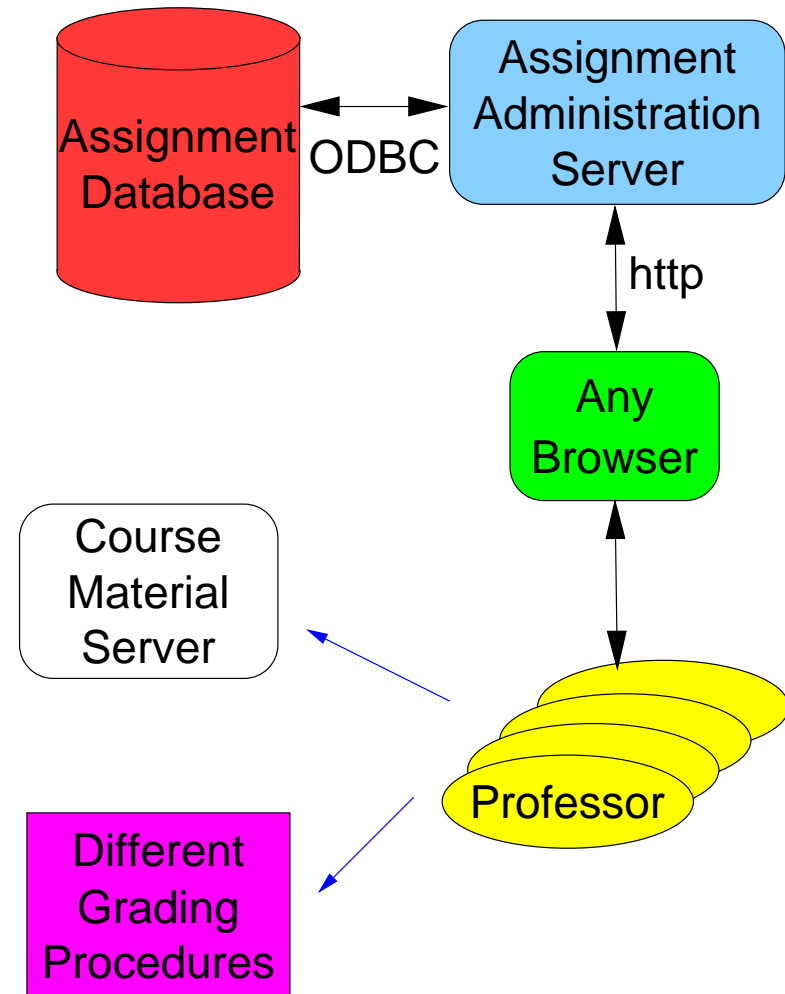
Student's View

- Interaction is done through a **browser**.
- Assignment available in a **regular web page**.
- Submission done through a **form** in an HTML page.
- Scores are published or sent **personally** to each student.



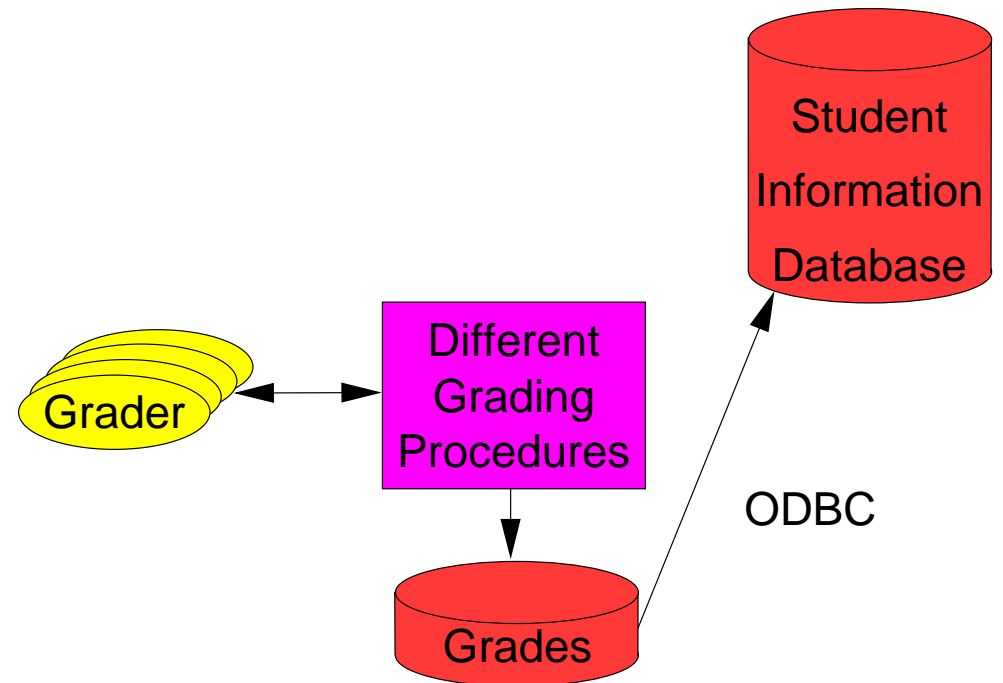
Professor's View

- Interaction through a browser with the **Assignment Database**.
- Provide the **administrative information**.
 - **Submission Period.**
 - **Authentication Database.**
 - **Sequence of Automatic Grading Procedures.**
- Eventually design the **Grading Procedures**.
- Design and publish the **Assignment Material**.



Grader's View

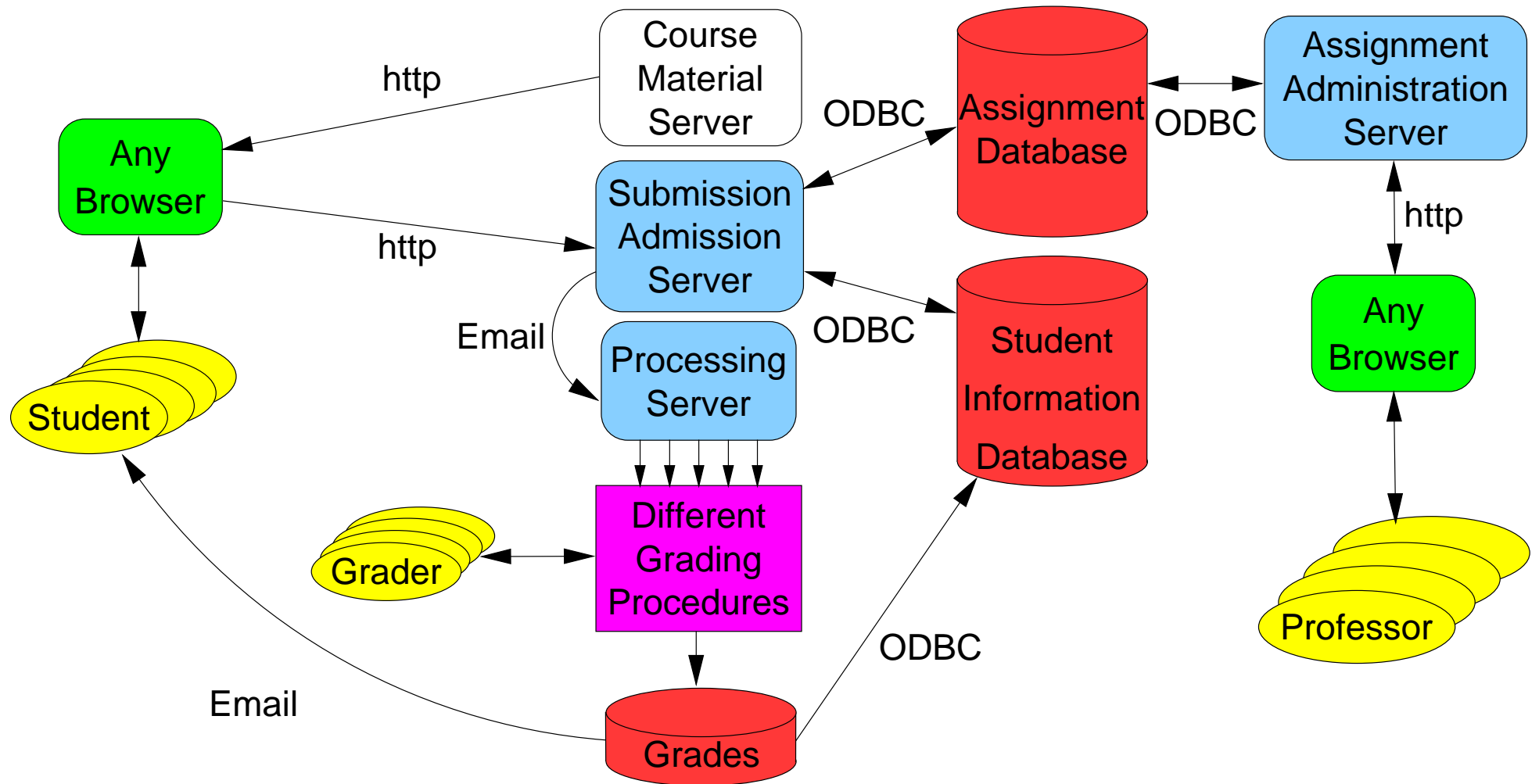
- Interaction with the submitted data.
- Use of conventional XML editors.
- Grades introduced in predefined format.



Multi-Agent Architecture

- **Assignment Administration Server:** Handles administrative information and accesses remote assignment database.
 - Several instances of this server may share a common remote database.
- **Submission Server:** Responsible for authentication and administrative checking.
 - Submission can be handled by several servers. They must share access to the assignment and student databases.
- **Processing Server:** Stores submitted data and processing procedures.
 - Grading can be divided among several processing servers. The grading procedures must be replicated.

Platform Architecture



Outline

- Introduction
- Related Research
- Overview of the Proposed Architecture
- The Student, Professor and Grader view
- **Sample Project**
- Conclusions and Future Work

Example Project

- Let us consider a conventional programming assignment with **peer review**.
- Students **submit** their work.
- Students **review** a peer's code.
- Students **modify** their code according to reviews.
- Procedure **repeats two times**.

Example Project

- Process requires **five submissions**. Five entries in the assignment database with the appropriate administrative information.

1. Initial code submission.

2. First review submission.

3. Second code submission.

4. Second review submission.

5. Final code submission.

Example Project: Processing Instructions

- **Processing Instructions for the first submission:**
 - **Divide the submissions in pairs to exchange reviews and store these pairs in an XML document.**
 - **Send received code to the appropriate reviewers.**
- **Processing Instructions for the second submission:**
 - **Receive reviews and send them to the appropriate authors.**
 - **Check for reviews not received.**
- **Processing Instructions for the third and fourth submissions are similar to the first and second.**
- **Final code submission is available to the graders.**

Additional Grading Procedures

- Automatic publishing of received submissions.
- Automatic test-bench execution and result notification.
- Automatic grade publishing.
- Automatic individual email notification (includes personalized comments).
- Procedures mostly implemented using XSL transformations and Java.

Hardware/Software Requirements

- **Hardware:** Computer(s) capable of hosting a web server with dynamic page generation and/or a database server.
- **Software:**
 - Javatm Runtime Environment.
 - Apache HTTP Server.
 - JavaServer Pages.
 - SQL Database Server (e.g. PostgreSQL).
 - XSL Processor (e.g. Xalan).
 - Email processing agent (e.g. procmail).

Current Status

- + First version already being used in a variety of courses in UC3M.
- + Used by more than 200 students and 5 faculty members.
- + Drastically reduced administrative load.
- + Close to a publicly available release.
- Linux based. Trivial migration to Windows in all servers except the processing server.

Outline

- Introduction
- Related Research
- Overview of the Proposed Architecture
- The Student, Professor and Grader view
- Sample Project
- **Conclusions and Future Work**

Conclusions

- A platform for **automatic assessment processing** has been presented.
- Focus on assessment requiring **human intervention**.
- Interface designed for **ease of use**.
- Technical proficiency kept to a minimum **except for the grading procedures**.
- **Reduced Hardware/Software requirements**.
- **Scalable to large classes**.
- **Open platform** to facilitate integration with other educational support tools.

Future Work

- Increase the default **grading procedures**.
- Simplify installation procedure and produce a **publicly available release**.
- Extend the system to include some sort of **fully automated grading**.
- Provide support for **adaptive evaluation**.

Thank you for your attention