

# Prototypische Integration automatisierter Programmbewertung in das LMS Moodle

Sebastian Becker, Andreas Stöcker, Daniel Bräckelmann,  
Robert Garmann, Sören Grzanna, Felix Heine, Carsten Kleiner,  
Peter Werner, Oliver J. Bott

Hochschule Hannover

APB 2013 Workshop "Automatische Bewertung von  
Programmieraufgaben"

# Gliederung

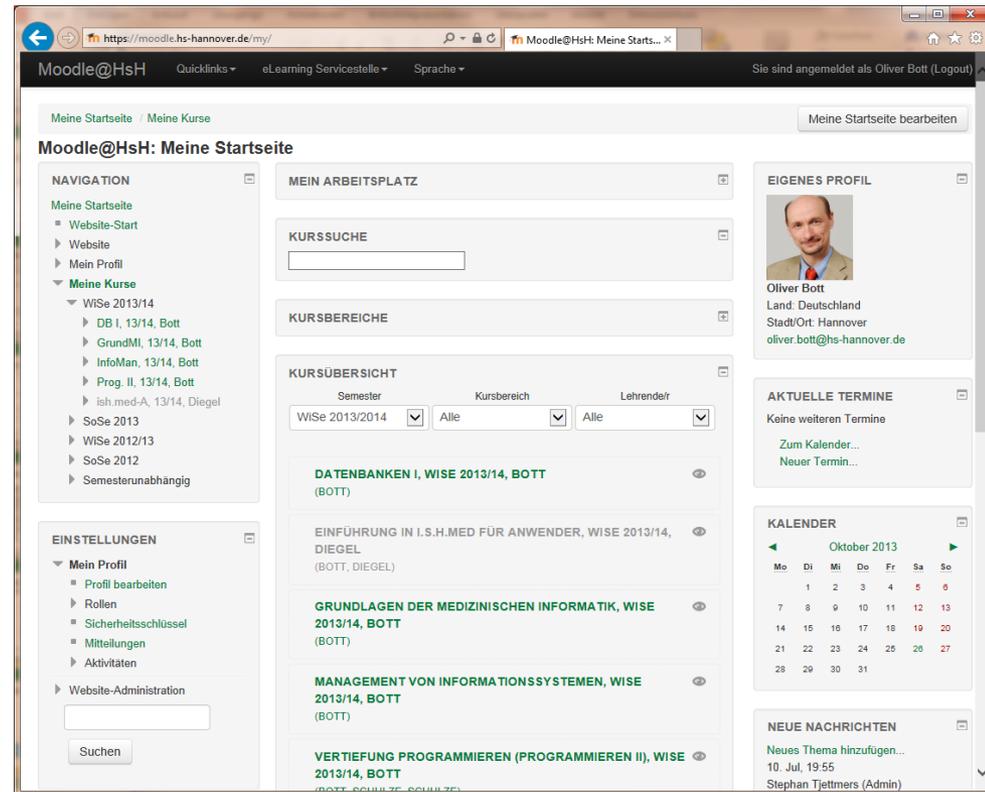
- Hintergrund und Fragestellung
- Basissysteme
  - Das LMS Moodle
  - Die Autograder aSQLg und Graja
- Die Middleware Grappa
  - Konfiguration des Grappa-Servers
  - Ablauf (Setup-Schnittstelle)
  - Grappa-Schnittstelle für Grader
  - Ablauf-Übersicht für aSQLg mit Einsatz von Grappa
- Moodle-Integration
  - Anlage einer Aufgabe durch den Dozenten
  - Ergebnis-Anzeige in Moodle aus studentischer Perspektive
- Ausblick / Diskussion

# Hintergrund und Fragestellung

- **Hintergrund:**
  - Studentischer Programmcode soll automatisiert überprüft werden
  - Mittlerweile viele Grader vorhanden, die idealerweise in die jeweils vorhandenen LMS eingebunden werden sollten
- **Problem:** Einbindung in LMS ist sehr aufwendig
  - Integration muss jedes Mal individuell programmiert werden
  - einheitliche Schnittstellen sind nicht vorhanden
- **Idee:** Entwicklung einer Middleware => Grappa-Webservice (= Grader Wrapper)
- **Test** anhand LMS Moodle und Grader aSQLg und Graja
  - Moodle ist LMS an der HsH
  - Graja und aSQLg sind an der HsH entwickelte Grader

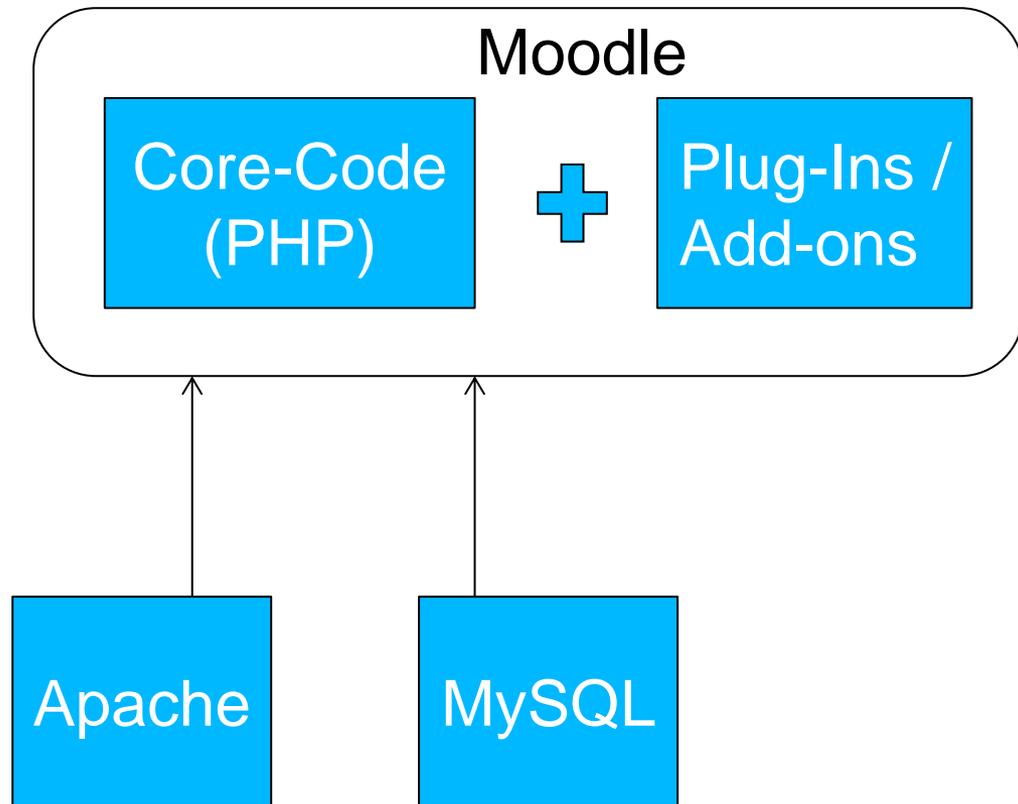
# Basissysteme: Das LMS Moodle

- Bereitstellung von Lerninhalten (Folien, Skripte, Videos, etc.)
- Organisation von Lehrveranstaltungen, Laborversuchen, etc.
- Kommunikationsunterstützung
- Lernzielkontrollen (Übungsaufgaben und Tests)
- Austauschmöglichkeit über Foren und Wiki
- Etc.



# Basissysteme: Das LMS Moodle

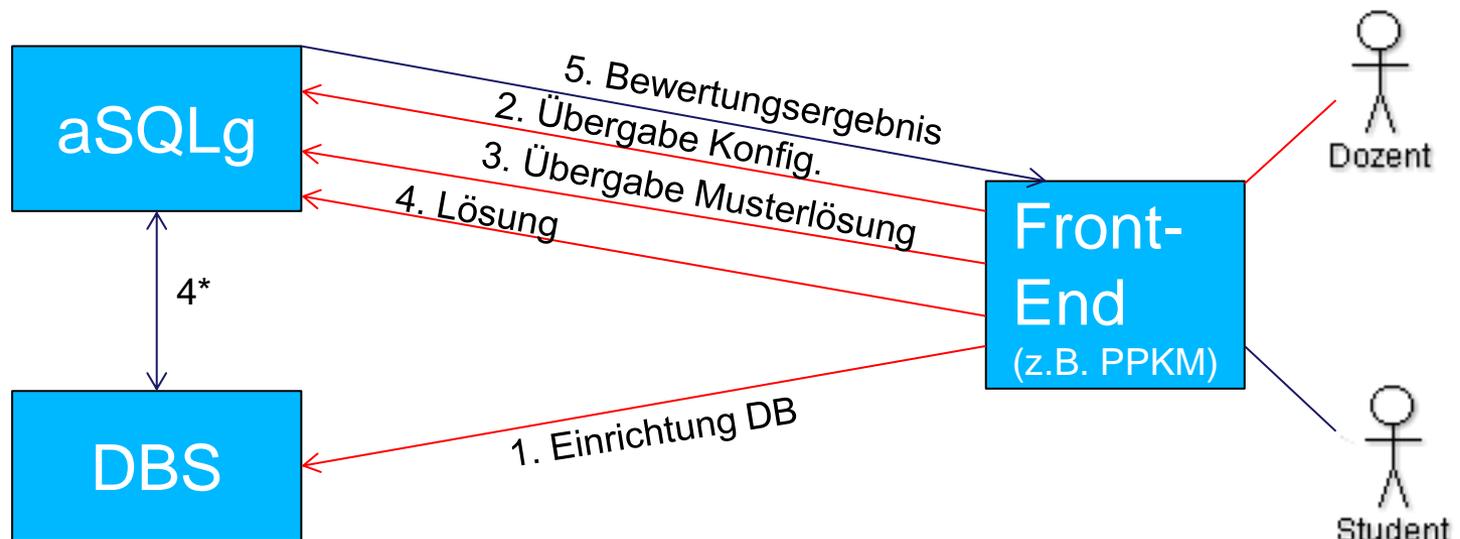
## Systemarchitektur von Moodle



Plugins may be either standard plugins (included in the Moodle download package) or add-ons.

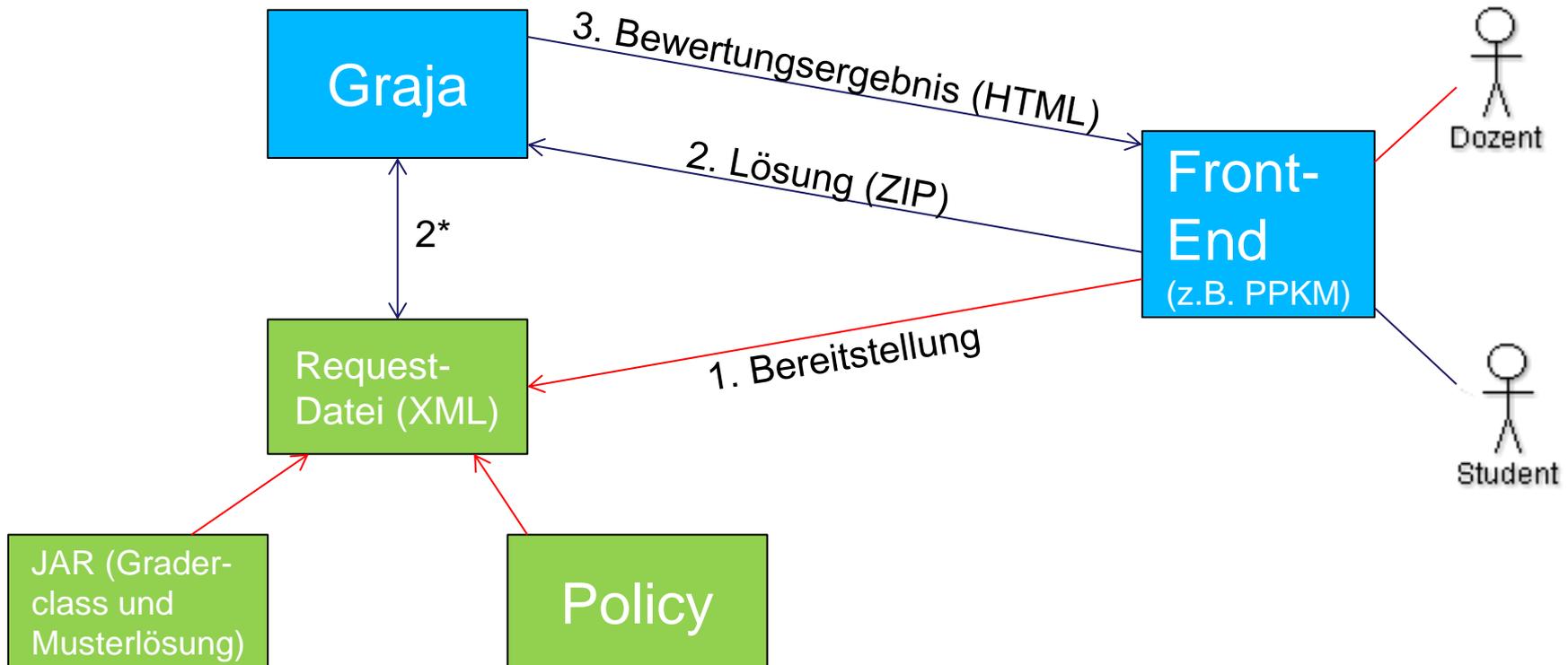
# Basissysteme: Der Autograder aSQLg

## Systeminteraktion ohne Grappa

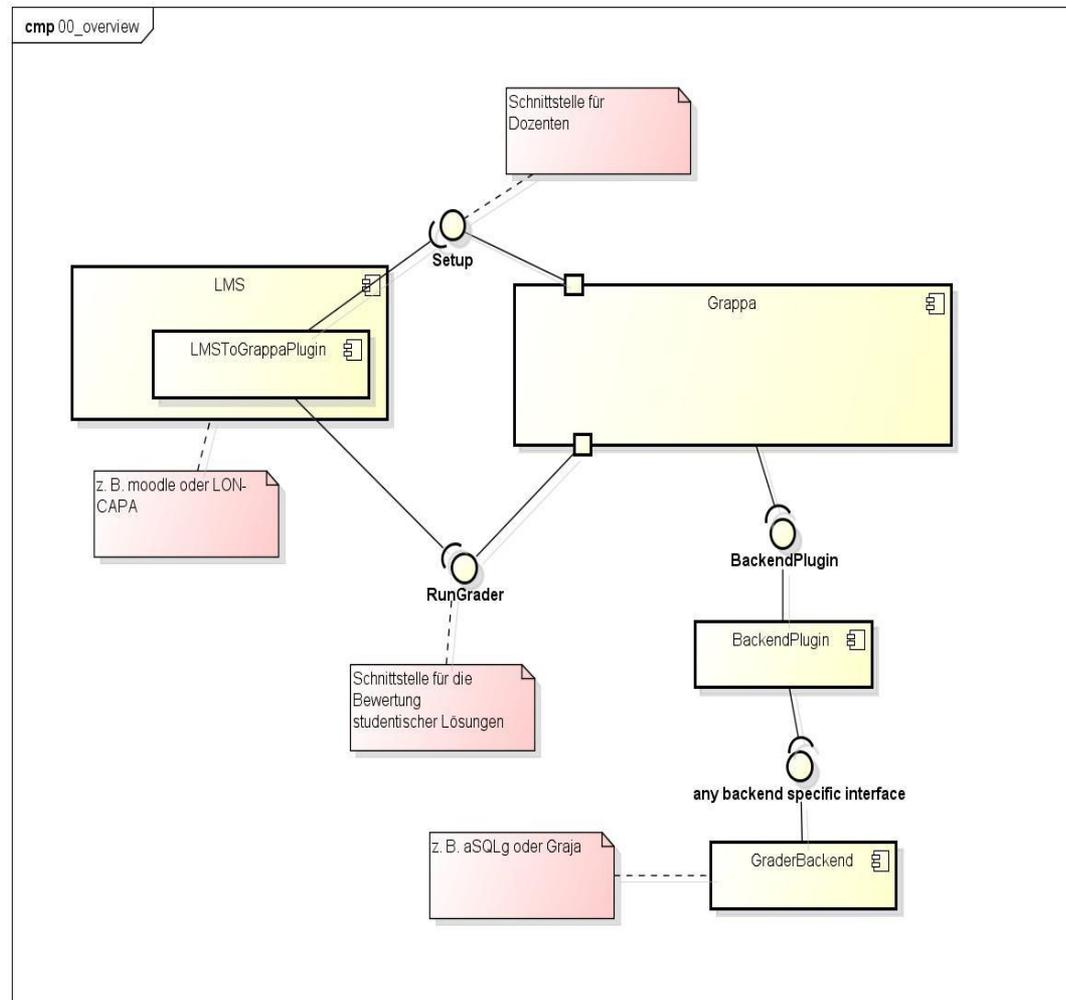


# Basissysteme: Der Autograder Graja

## Systeminteraktion ohne Grappa

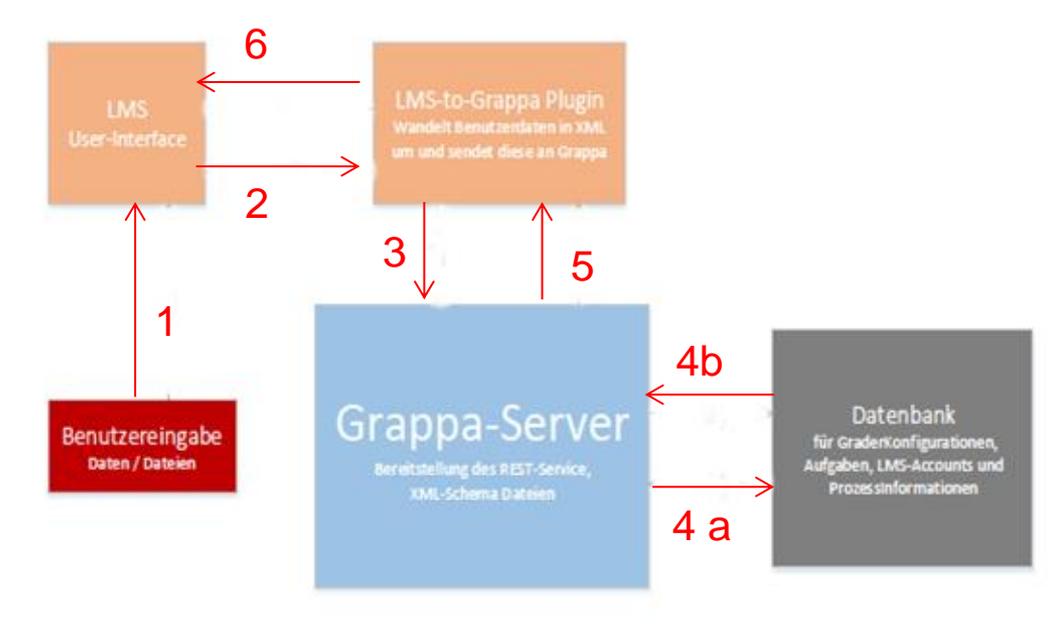


# Die Middleware Grappa



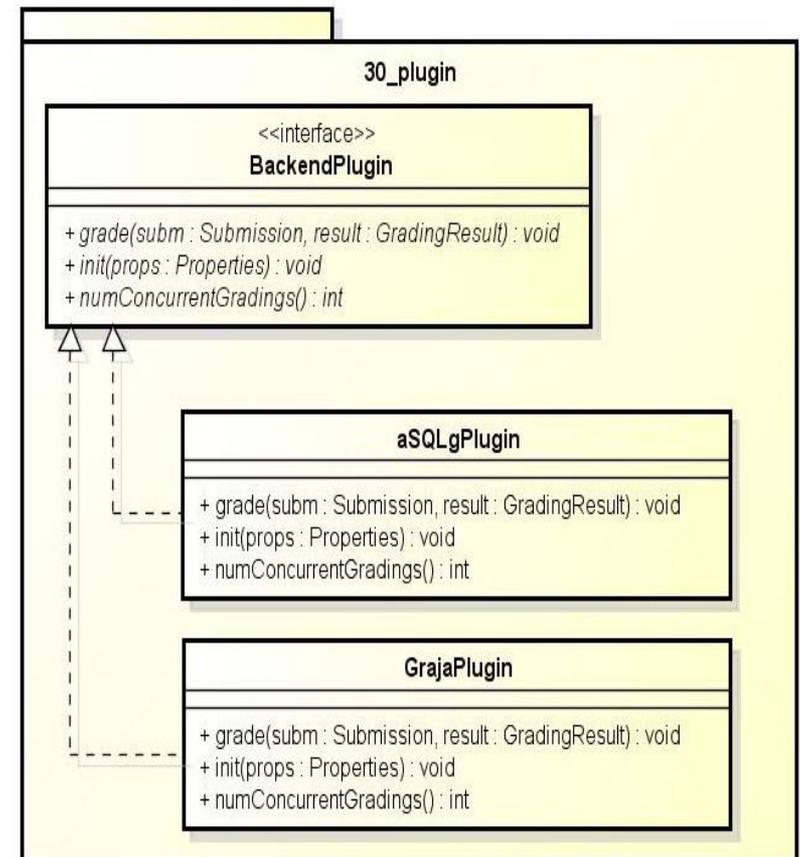
# Ablauf anhand der Setup-Schnittstelle

- 1) Dozent stellt dem LMS die benötigten grader-individuellen Konfigurations-Daten bzw. -Dateien zur Verfügung
- 2) Die Konfigurationsdaten werden an das Plug-In übermittelt. Dort werden sie in Schemakonforme XML-Tags gewandelt.
- 3) Das Plug-In überträgt per HTTP-Request die erstellte XML-Datei
- 4) Der Grappa-Server wertet die Informationen von LMS-Account und XML-Datei aus:
  - 1) Speicherung in der Datenbank falls nicht vorhanden
  - 2) Fehlermeldung falls bereits vorhandene ID
- 5) HTTP Response „OK“ oder entsprechende Fehlermeldung
- 6) Übermittlung und Verarbeitung der Status Informationen (geplant)



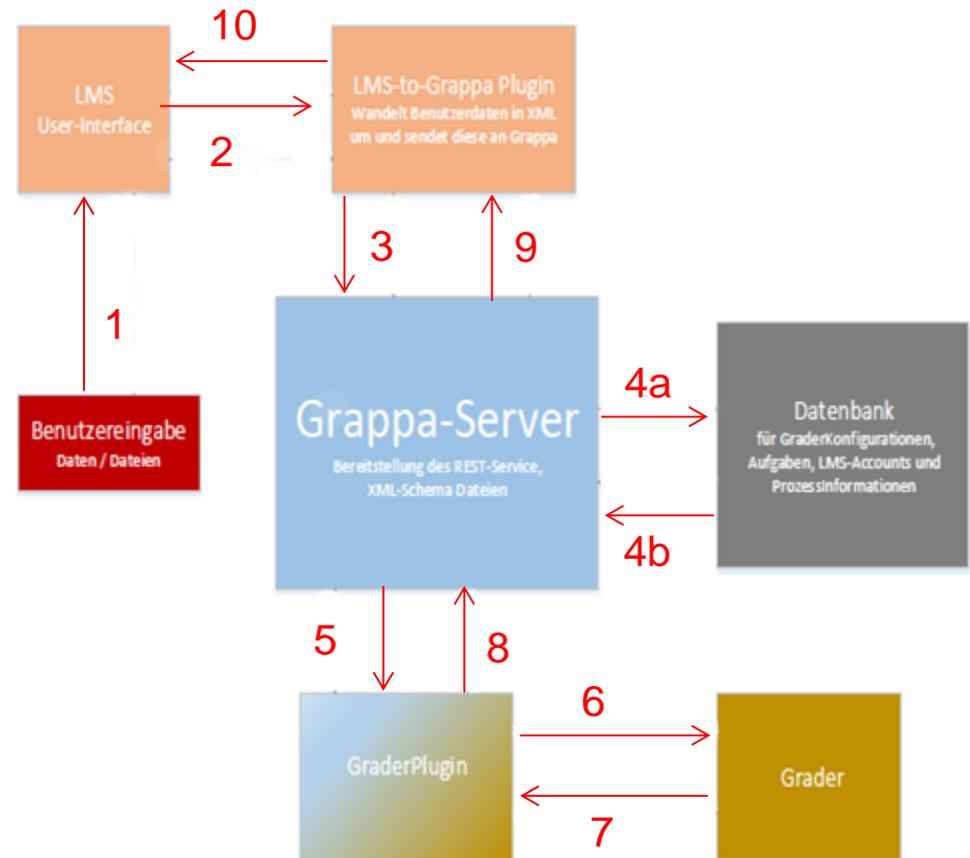
# Grappa-Schnittstelle für Grader

- Jedes Graderspezifische Plug-In muss das Interface implementieren
- Per *init()* können Graderabhängige Properties übergeben werden
- Für den *grade()*-Aufruf muss Grappa ein *GradingResult*-Objekt erstellen
- *numConcurrentGradings* ist die maximale Anzahl von gleichzeitigen *grade()*-Aufrufen (derzeit noch nicht implementiert)

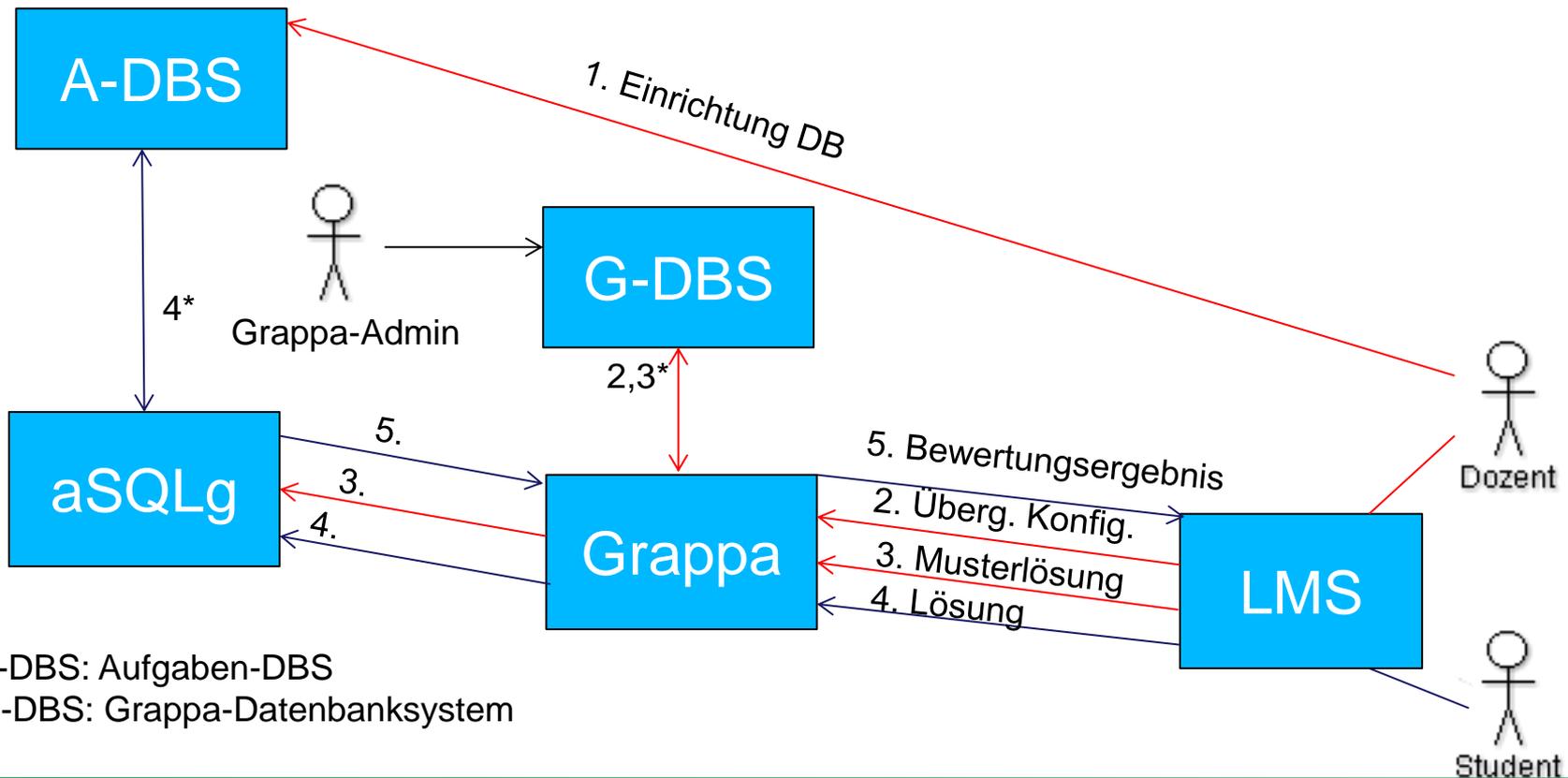


# Ablauf anhand der Grader-Schnittstelle

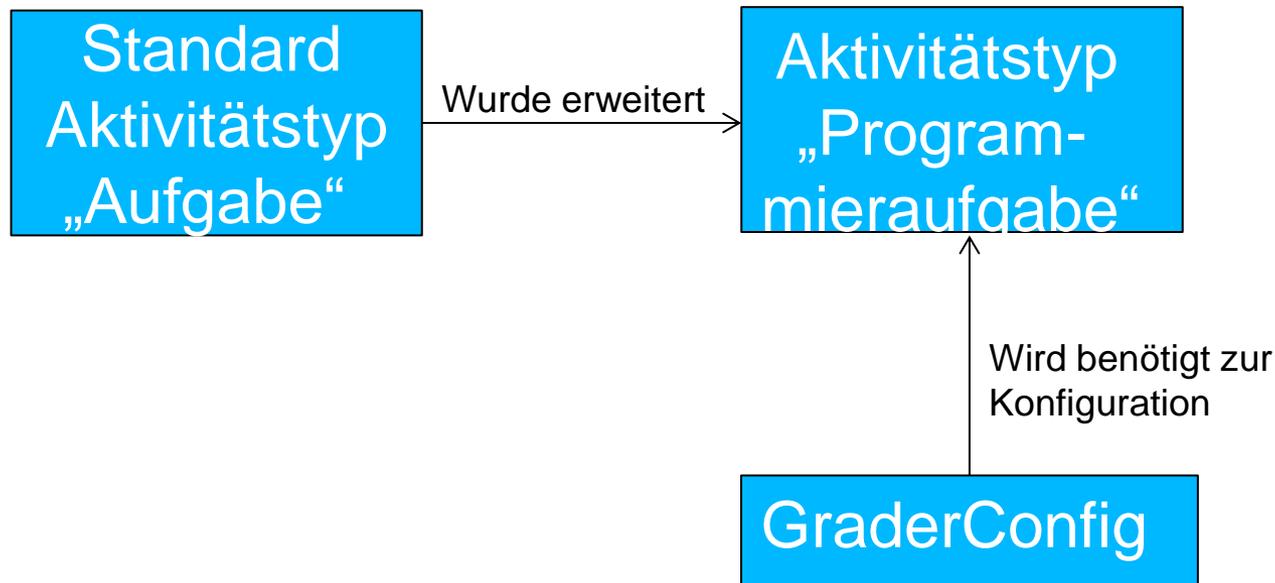
- 1) Student stellt dem LMS die benötigten Daten und ggf. Dateien zur Verfügung
- 2) Die Formulardaten werden dem Plug-In zur Verfügung gestellt. Dort werden sie in Schema-entsprechende XML-Tags umgewandelt.
- 3) Das Plug-In überträgt per HTTP Request die erstellte XML-Datei
- 4) Der Grappa-Server wertet die Informationen von LMS-Account und XML-Datei aus
  - 1) Anforderung der Aufgabe anhand der ID und LMS
  - 2) Rückgabe der Aufgabe inkl. der Grader-Konfigurationen
- 5) Evtl. Konfiguration des Grader-Plug-Ins und Bereitstellung eines aus der Aufgabe erstellten Result-Objekts und der studentischen Lösung
- 6) Umwandlung der Grappa-spezifischen Daten in Grader-spezifische Objekte, evtl. Konfigurationen etc. und starten des Grade-Vorgangs
- 7) Resultat des Graders oder Fehlermeldung
- 8) Transformation des Grader-spezifischen Resultats oder Fehlers für Grappa und Rückgabe
- 9) Validierung des Resultats mit anschließender Umformung in XML-Daten
- 10) XML-Daten auswerten und graphisch darstellen



# Ablauf-Übersicht zu aSQLg mit Grappa-Einsatz



# Programmierung des Moodle-Plug-Ins



Enthält Individuelle Graderplugin-Einstellungen für jeden Gradertyp, wie z.B. Ergebnisformat, Datei- oder Textformat für Lösungsübergabe,...

## Grappa Test Kurs

**NAVIGATION**

- Meine Startseite
  - Website-Start
  - Website
  - Mein Profil
  - Dieser Kurs
    - gtk
      - Teilnehmer/innen
      - Allgemeines
      - Erste Schritte in SQL

**SUCHE IN FOREN**

Erweiterte Suche

**Erste Schritte in SQL**

- Aufgabe 1
  - a1\_solution\_id15

**AKTIVITÄTEN**

- Abstimmung
- Aufgabe
- Chat
- Datenbank
- Externes Tool
- Feedback
- Forum
- Glossar
- Journal
- Lektion
- Programmieraufgabe**
- Test
- Wiki

**ARBEITSMATERIAL**

- Buch

Stellen Sie für die Teilnehmer/innen Ihres Kurses Programmieraufgaben bereit, die diese online oder offline bearbeiten. Die Lösung kann online als Texteingabe oder Dateiabgabe erfolgen. Geben Sie konstruktiv Feedback und bewerten Sie die Lösung.

Bewertungsmöglichkeiten: - mit vorgegebener oder selbst erstellter Bewertungsskala - durch Text-Feedback - mit Dateianhang (z.B. Musterlösung oder korrigierte Lösung) - Bewertung mit mehreren Kriterien.

[Weitere Hilfe](#)

Hinzufügen Abbrechen

# Moodle-Integration: Anlage einer Aufgabe durch den Dozenten

# Moodle-Integration: Ergebnis-Anzeige aus studentischer Perspektive

https://moodev1.hs-hannover.de/mod/grappa/view.php?id=17620

Moodle@HsH Quicklinks eLearning Servicestelle Sprache Sie sind angemeldet als Sören Grzanna (Logout)

Feedback comments

| Aufgabe         | Punkte    | Prüfungen  |             |        |            |               |           |                         |               |           |  |                 |           |   |
|-----------------|-----------|--|-------------|--------|------------|---------------|-----------|-------------------------|---------------|-----------|--|-----------------|-----------|---|
| 01              | 1.0/1.0   | <table border="1"><thead><tr><th>Prüfungstyp</th><th>Punkte</th><th>Kommentare</th></tr></thead><tbody><tr><td>Syntaxprüfung</td><td>0.33/0.33</td><td>Die Syntax ist korrekt.</td></tr><tr><td>Kostenprüfung</td><td>0.33/0.33</td><td>Kosten Ihres Statements: 3.0.<br/>Absolute Kostenobergrenze ist: 100000.<br/>Kosten der Musterlösung: 3.0.<br/>Die Kosten der Abfrage sollte kleiner als 6.0 sein.<br/>Die Kosten der Abfrage sind ok.</td></tr><tr><td>Ergebnisprüfung</td><td>0.33/0.33</td><td>Die Spaltenanzahl Ihres Ergebnisses ist richtig.<br/>Die Datentypen der Spalten Ihres Ergebnisses sind richtig.<br/>Die Namen der Spalten Ihres Ergebnisses sind richtig.<br/>Ihr Ergebnis hat die richtige Zeilenanzahl.<br/>Die Werte Ihrer Ergebniszeilen sind richtig.<br/>Das Ergebnis ist vollständig richtig.</td></tr></tbody></table> | Prüfungstyp | Punkte | Kommentare | Syntaxprüfung | 0.33/0.33 | Die Syntax ist korrekt. | Kostenprüfung | 0.33/0.33 | Kosten Ihres Statements: 3.0.<br>Absolute Kostenobergrenze ist: 100000.<br>Kosten der Musterlösung: 3.0.<br>Die Kosten der Abfrage sollte kleiner als 6.0 sein.<br>Die Kosten der Abfrage sind ok. | Ergebnisprüfung | 0.33/0.33 | Die Spaltenanzahl Ihres Ergebnisses ist richtig.<br>Die Datentypen der Spalten Ihres Ergebnisses sind richtig.<br>Die Namen der Spalten Ihres Ergebnisses sind richtig.<br>Ihr Ergebnis hat die richtige Zeilenanzahl.<br>Die Werte Ihrer Ergebniszeilen sind richtig.<br>Das Ergebnis ist vollständig richtig. |
| Prüfungstyp     | Punkte    | Kommentare   |             |        |            |               |           |                         |               |           |  |                 |           |   |
| Syntaxprüfung   | 0.33/0.33 | Die Syntax ist korrekt.  |             |        |            |               |           |                         |               |           |  |                 |           |   |
| Kostenprüfung   | 0.33/0.33 | Kosten Ihres Statements: 3.0.<br>Absolute Kostenobergrenze ist: 100000.<br>Kosten der Musterlösung: 3.0.<br>Die Kosten der Abfrage sollte kleiner als 6.0 sein.<br>Die Kosten der Abfrage sind ok.   |             |        |            |               |           |                         |               |           |  |                 |           |   |
| Ergebnisprüfung | 0.33/0.33 | Die Spaltenanzahl Ihres Ergebnisses ist richtig.<br>Die Datentypen der Spalten Ihres Ergebnisses sind richtig.<br>Die Namen der Spalten Ihres Ergebnisses sind richtig.<br>Ihr Ergebnis hat die richtige Zeilenanzahl.<br>Die Werte Ihrer Ergebniszeilen sind richtig.<br>Das Ergebnis ist vollständig richtig.  |             |        |            |               |           |                         |               |           |  |                 |           |   |

# Ausblick / Diskussion

- Bald auch Tests in Moodle möglich durch Erweiterung des Aktivitätstyps „Test“
- Export- und Importfunktion in Moodle für den Aufgabenaustausch (ProFormA AG XML-Format)
- Einlesen des Repositories innerhalb von Moodle (Suche darin)
- Durch Verwendung der REST Schnittstelle-auch Verwendung von Grappa innerhalb anderer LMS (=> LON-CAPA)
- Aktuell wird das System in einem DB-I-Kurs evaluiert

# Danke für Ihre Aufmerksamkeit

Kontakt:

Sebastian Becker

E-Learning Servicestelle

Email: [sebastian.becker@hs-hannover.de](mailto:sebastian.becker@hs-hannover.de)