PePP
Partnerschaft für innovative E-Prüfungen
Projektverbund der baden-württembergischen Universitäten

Universität Stuttgart

Online-Programmierung mit ViPLab und Jupyter

Admir Obralija[1] Ina Baret[1] Oliver König[2]

PePP-Netzwerktreffen 2024

11.06.2024

[1] Technische Informations- und Kommunikationsdienste der Universität Stuttgart
[2] Institut für Angewandte Analysis und Numerische Simulation -- Universität Stuttgart

»Partnerschaft für innovative E-Prüfungen. Projektverbund der baden-württembergischen Universitäten (PePP)«

universität freiburg · Universität Heidelberg · Universität Hohenheim · KIT Karlsruher Institut für Technologie · Universität Konstanz · Universität Mannheim · Universität Stuttgart · Eberhard Karls Universität Tübingen · universität ulm

gefördert von der »Stiftung Innovation in der Hochschullehre«

Stiftung Innovation in der Hochschullehre

PePP

Übersicht

- Virtuelle Programmierumgebungen
- ViPLab
- Reallabor und Dozierendenbefragung
- Jupyter
- Rollout und Repositories

- Demo: Prüfungsworkflow & Beispiele aus der Lehre
- Workshop: Interaktion mit virtuellen Programmierumgebungen

PePP

Motivation

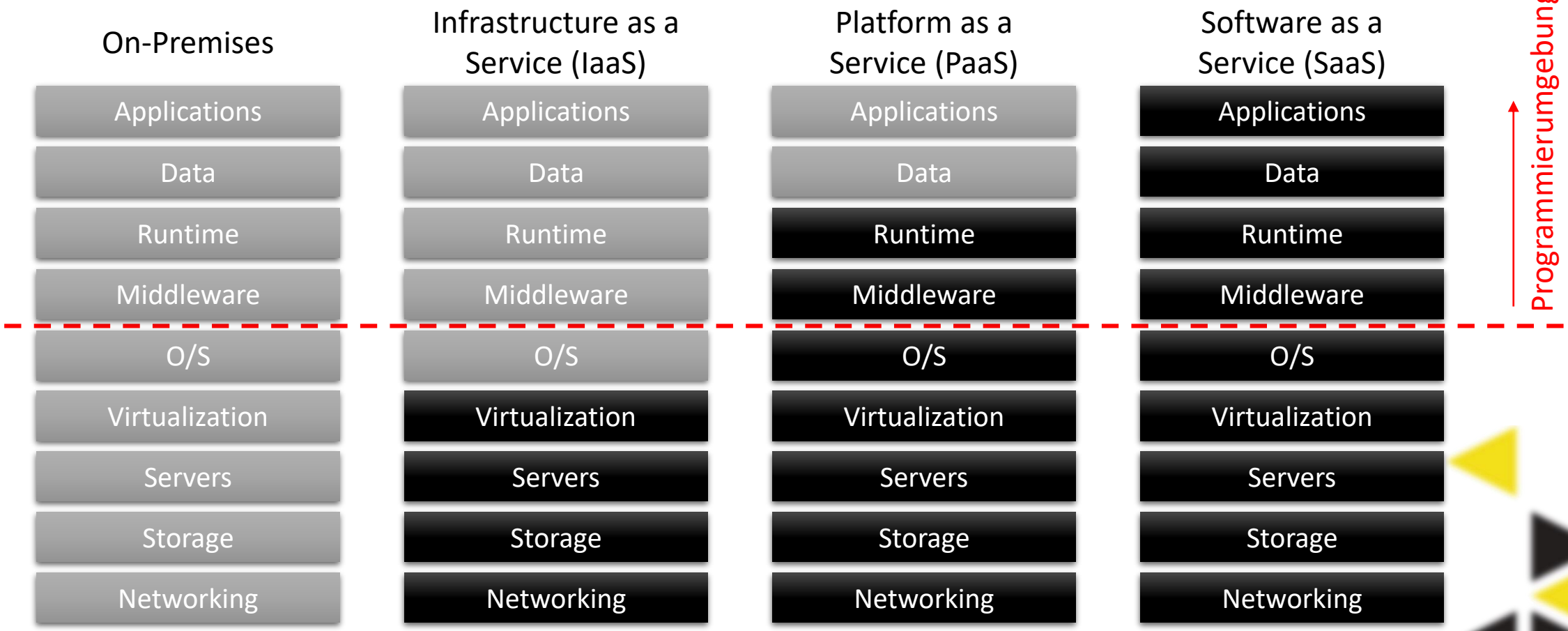
- Programmierung ist essentieller Bestandteil der Wissenschafts- und Ingenieursausbildung
- **Medienbruch**: Etablierter Übungsworkflow am Rechner, jedoch handschriftliche Programmierung in Prüfungen
- Studierende lernen die Prüfungsumgebung im Übungsbetrieb kennen
- Zielgruppe: Studienanfänger*innen und Fortgeschrittene
  - Grundlagenkurse (z.B. Numerik, Algorithmik, Modellierung, etc.)
  - Spezialisierungsmodule (z.B. Data Science, Machine Learning, etc.)
- Motivation: **Einheitliche Übungs- und Prüfungsumgebung ohne Mehraufwand**

Programmierumgebungen und ihre Typen

- Code- und Texteditoren mit Syntax-Unterstützung (z.B. vim, emacs, notepad++, etc.)
  - Keine Compiler oder Interpreter
- Integrierte Entwicklungsumgebungen (IDEs, z.B. IntelliJ, Webstorm, Eclipse, Codeblocks, etc.)
  - Compiler, Debugger, Profiler, Versionskontrolle
  - Ausführung des Programms nach Kompilierung oder Intepretation
    - Ahead-of-time- vs. Runtime-Kompilierung
- **Virtuelle und interaktive Programmierumgebungen**
  - Vorteile der IDEs, jedoch keine ausführbaren Programme für den Produktionsbetrieb
  - Cloud-basiert

PePP

Programmierumgebungen und Cloud



PePP

Gegenüberstellung

Pen & Paper

- ✗ Keine Kompilierung und Visualisierung der Ergebnisse
- ✗ Kein unmittelbares Feedback (Compilerausgaben, Fehler, etc.)
- ✗ Kein reeller Kontext
- ✗ Korrekturaufwand
- ✓ Ausfallsicherheit

Kursspezifische Bereitstellung durch Lehrende

- ✗ Hoher konfigurativer und organisatorischer Aufwand
- ✗ Kein Schreibschutz- und Abgabesystem
- ✓ Flexibilität
- ✓ Reeller Kontext

Virtuelle Programmierumgebung

- ✓ Browser- und OS-unabhängig → BYOD
- ✓ Einheitliche Umgebung für Übungs- und Prüfungsbetrieb
- ✓ Automatisierte Bewertung
- ✓ Ein Arbeitsbereich
- ✓ Individuelle Compiler-Konfiguration pro Aufgabe
- ✓ In ILIAS integriert: Vorteile der Test & Assessment Features

PePP

Das Virtuelle Programmierlabor (ViPLab)

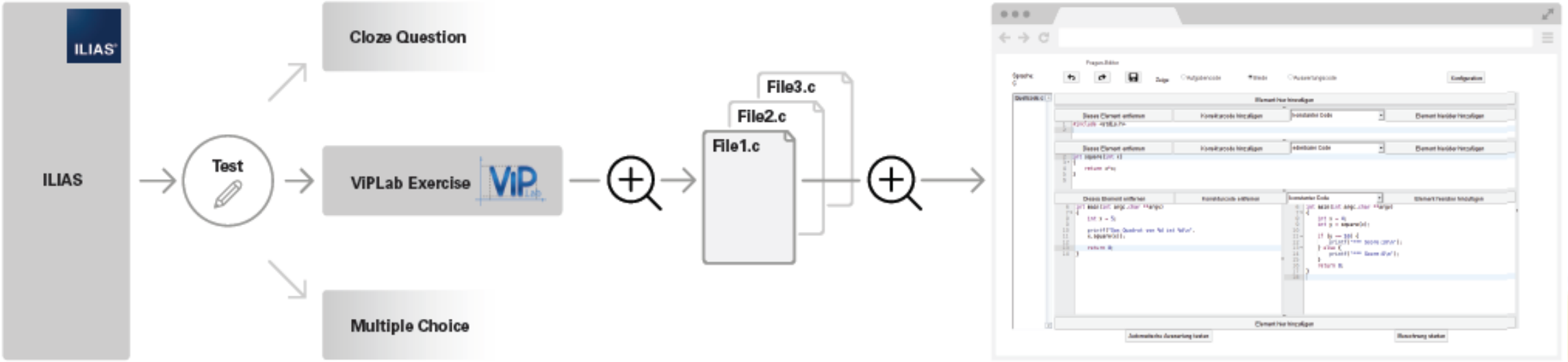
- Ein institutsübergreifendes Projekt an der Universität Stuttgart (seit 2009)
- Ursprünglich: Bereitstellung und Betreuung von Programmieraufgaben für große Numerik-Vorlesungen
- Start mit Unterstützung der Studierenden (Finanziert aus Studiengebühren)
- Seit 2010 in Übungen eingesetzt
- Richtige Bearbeitung der Aufgabenblätter → Bonuspunkte für die Prüfung
- **<u>Ziel</u>: Kombination von theoretischen Aufgaben und praktischen Programmieraufgaben**

Nachklausur

PePP

ILIAS-Integration

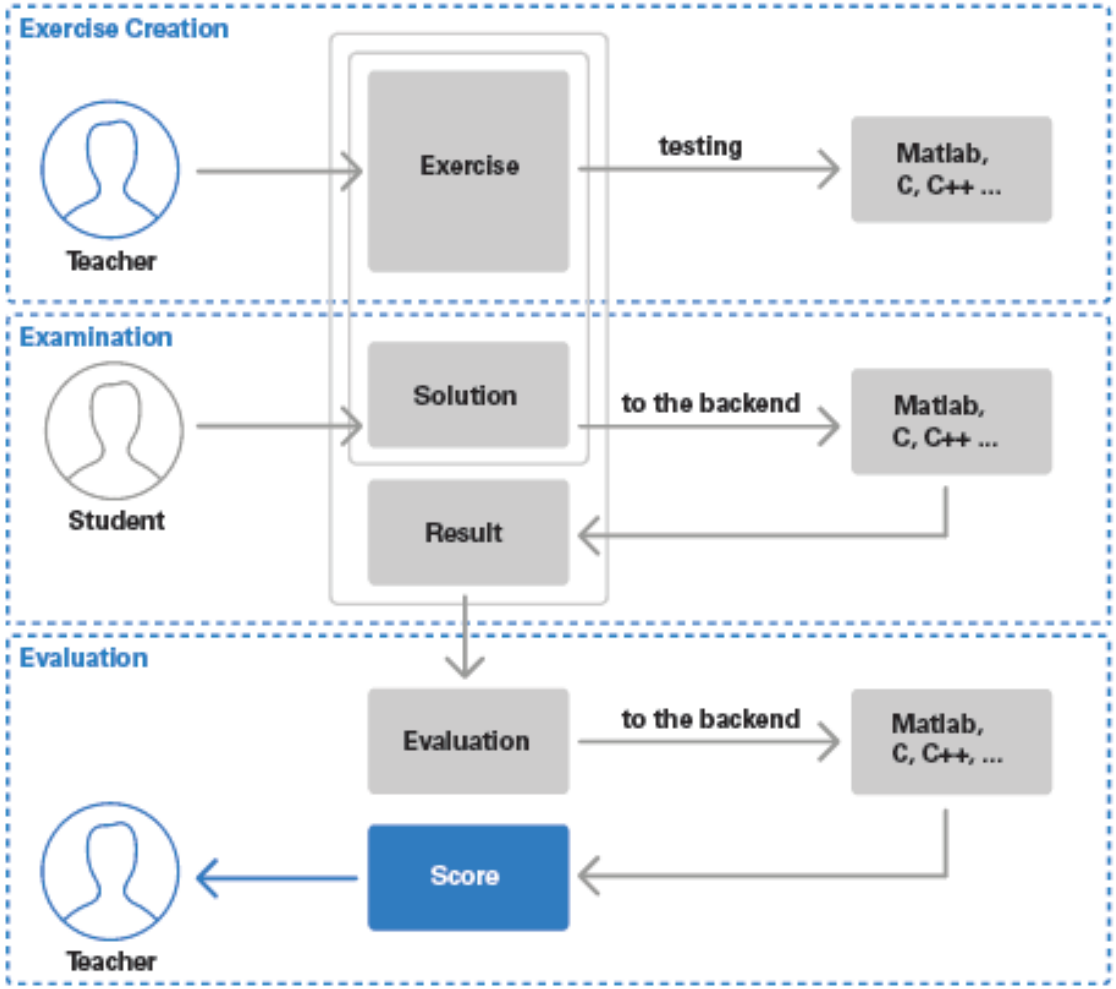
- Einheitliche Übungs- und Prüfungsumgebung für Dozierende und Studierende → ***Eintrainierter Workflow***
- Niederschwelliger Zugang via Webbrowser; d.h. Webbrowser-, OS- und Geräte-Unabhängigkeit
- Nutzung der bestehenden Test and Assessment Features aus ILIAS
- Kombination mit anderen Fragetyen



PePP

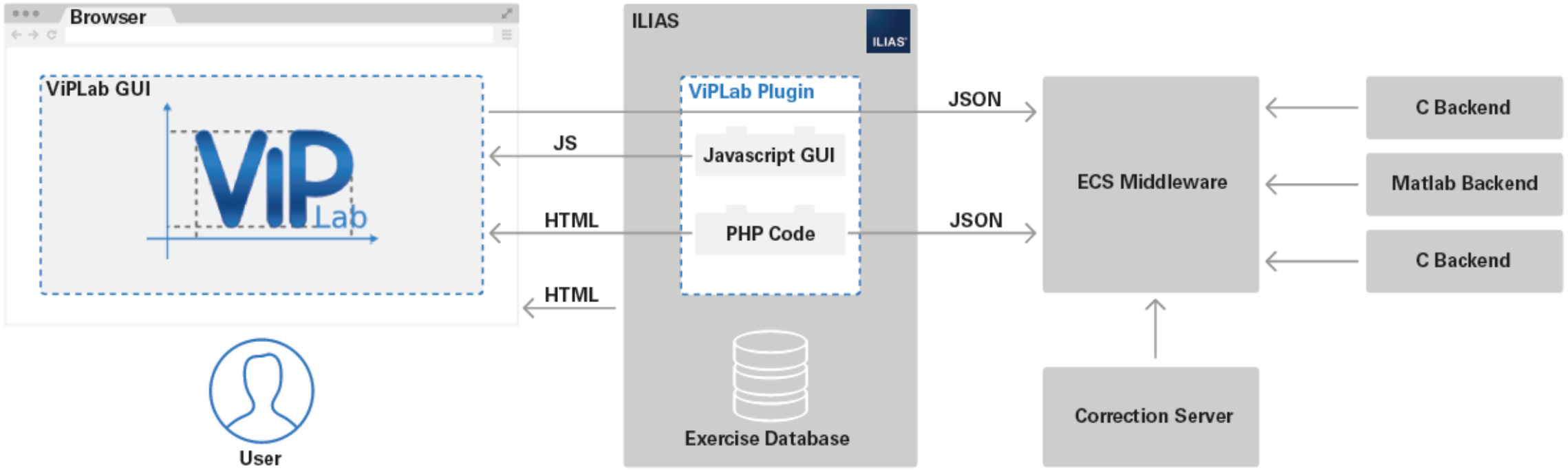
Prüfungsworkflow im Überblick

- Erstellung und Bestimmung der Sichtbarkeit/Änderbarkeit des Codes
- Kompilierung, Programmausgabe und Visualisierung
- Lösen durch Studierende
- Automatisierte Bewertung studentischer Abgaben *oder*
- Manuelle Bewertung anhand der Code-Abgabe und/oder Programmausgabe
- Feedback & Statistiken



PePP

ViPLab 2.0 – Systemumgebung und Architektur



PePP

Reallabor – Verlauf

…

Q4 2022

- Stellenbesetzung für Softwareentwicklung

Q1 2023

- Bugfixing und korrekte Inbetriebnahme des ViPLab 2.0-Plugins
- Stellenbesetzung für Nutzerbetreuung
- ViPLab 2.0-Importer und Exporter für ILIAS

Q2 2023

- Aufwandsschätzung ViPLab 3.0- / Jupyter-Integration
- Anleitungen

Q3 2023

- Grundfunktionalität Jupyter-Plugin für ILIAS
- Basiskonfiguration für Jupyterhub
- Leitfragen

Q4 2023

- 1. Iterationsphase der Jupyter-Entwicklung abgeschlossen
- Backend-Stack und skalierbare Jupyter-Infrastruktur
- Dozierendenbefragung

Q1 2024

- Umstellung der Ressourcenverwaltung nach ILIAS-IRSS
- Vorbereitungen für den Testbetrieb
- Publikation des Jupyter-Plugins für ILIAS
- Publikation der cloudfähigen Jupyterhub-Infrastrukturkonfiguration

…

PePP

Reallabor – Lehrendenbefragung

- Befragung Lehrende der Uni Stuttgart
- Ziel:
  - Grundsatzentscheidung Implementierung Jupyter/ViPLab 3
  - Bedarf Nutzung virtueller Programmierumgebungen
- Drei Kategorien von Fragen
  - Allgemeine Fragen: Format der Lehrveranstaltung, Studierendenanzahl
  - Didaktische Fragen: konkrete didaktische Einsatzszenarien der Programmierumgebung, Features
  - Technische Fragen: notwendige technische Infrastruktur

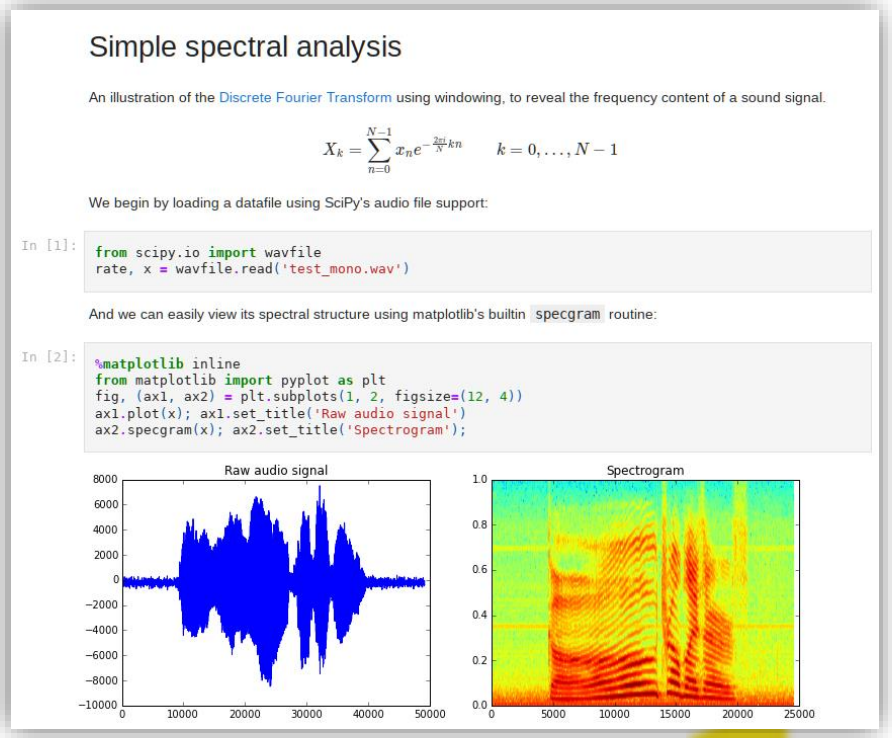
PePP

Lehrendenbefragung – Erkenntnisse

- Integration ILIAS Lernmodul oder Test
- Zuordnung Teilaufgaben (Lösungs-) Codesegment
- Formatieren Aufgabentext, z.B. mit LaTeX
- Real-Time-Kompilierung von Programmcode
- Verteilung der Aufgabenstellung über mehrere Dateien
- Eigene Umgebung (Programmiersprache/Compiler/Interpreter (Skriptsprachen))
- Abspeichern des Lösungscode neben Ausgabe, z.B. für Korrektur und Bewertung
- Hinzufügen von Hilfsdateien (z.B. Rohdateien)

Jupyter-Notebooks

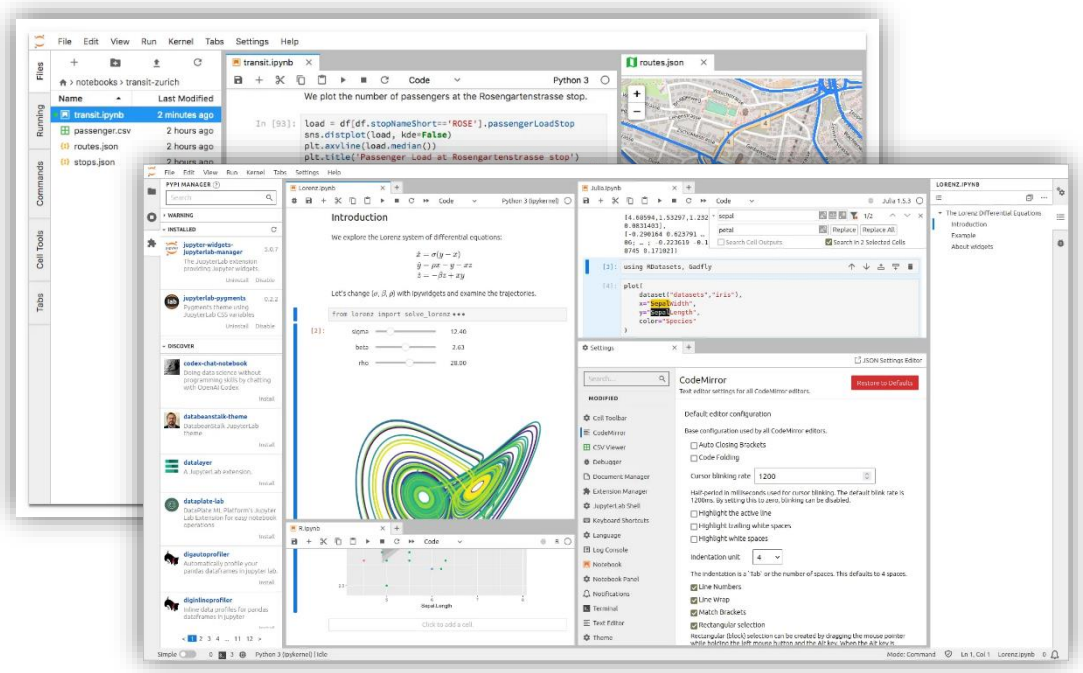
- Interaktive und literarische Programmierung
- Vollständige **Programmiersprachen** und **Bibliotheken** via **Jupyter-Kernel**
- Auszeichnungssprachen: Markdown, LaTeX, HTML, etc.
- Erweiterungen und Widgets
  - Beispiel: Parametrisierung über Steuerelemente
- JSON-basiertes Dateiformat
- Potentieller Einsatz in der Lehre und in E-Prüfungen
  - **Kontextsensitive** Verständnisabfrage innerhalb von Programmieraufgaben
  - Abfrage der **Benutzung** von speziellen Bibliotheken



[Grafik: github.com/jupyter]

PePP

JupyterLab

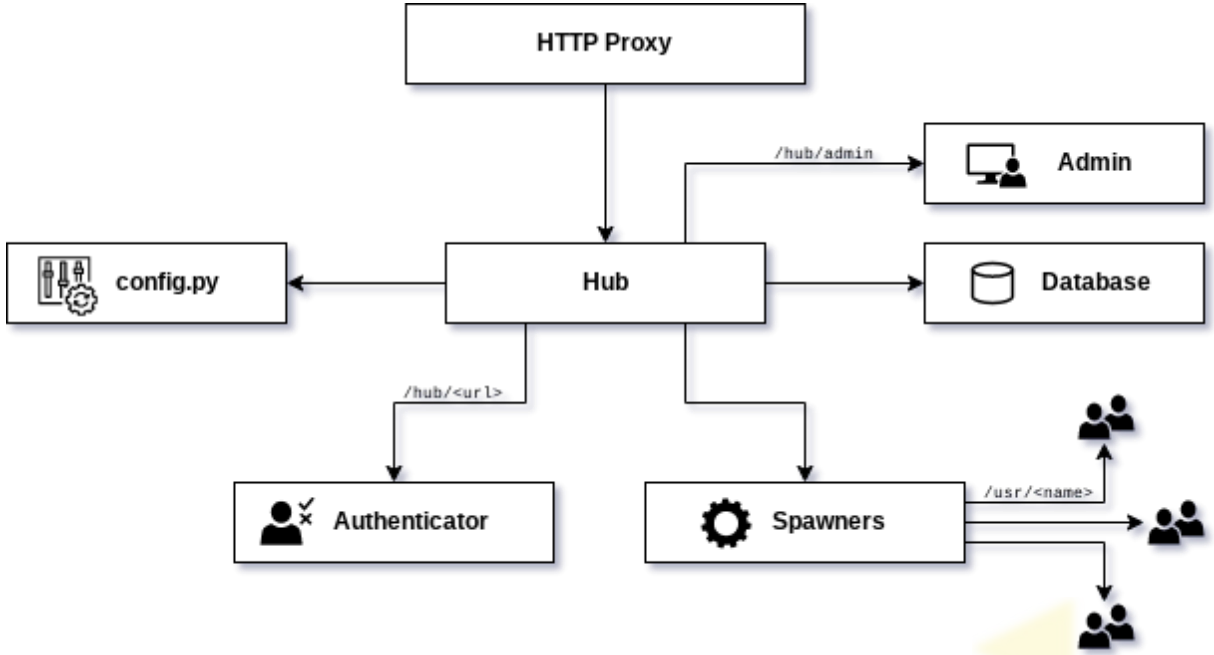


[Grafik: jupyter.org]

- Erhöhter Umfang des Arbeitsbereichs
  - Projekt-Explorer
  - Multi-window Ansicht
  - Textkonsolen
- Viewer für Grafiken, Rohdaten und Dokumente
- JSON-basiert
- Erweiterungen
  - Monitoring, VCS, Dashboards, etc.
  - Kollaboratives Arbeiten

Jupyterhub

- Cloud-fähige Mehrbenutzerumgebung
- API-Schnittstelle
- Authentifizierung und Generierung eines Nutzerservers
- Rekonfiguration des Proxy-Servers: Weiterleitung zum generierten Nutzerserver



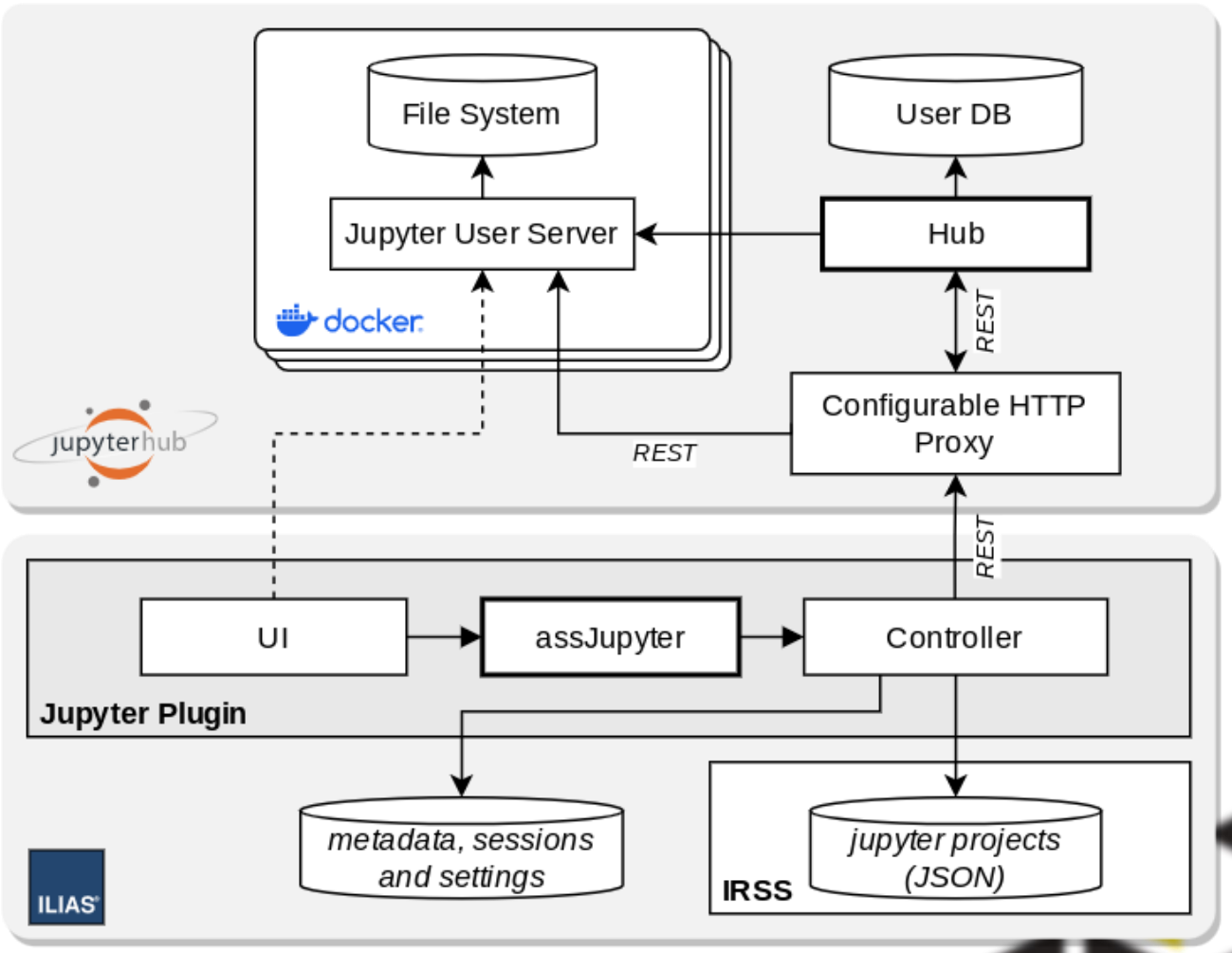
[Grafik: jupyterhub.readthedocs.io]

- Betrieb im Kubernetes-Cluster
  - Schnelles Hochfahren von temporären Jupyter-Sessions
  - Erfüllung spezieller Hardwareanforderungen, z. B. GPU-Tensorflow (Deep Learning Framework)

PePP

Jupyter-Plugin für ILIAS

- Verwendung bestehender REST-API-Schnittstellen → Minimalinvasive Integration
- ILIAS-interne Authentifizierung mittels API-Tokens
- UI-Integration via iframe
- ILIAS als führendes Datenhaltungssystem
- Exporter und Importer für Jupyter-Projekte
- Plugin äquivalent zur ViPLab-Integration



PePP

Jupyter-Plugin für ILIAS – Herausforderungen

- Session-Management und Synchronisation
- Cross-Origin Resource Sharing (CORS)
- Abbildung der Jupyterhub-Datenstruktur ins IRSS
- Abschirmung gegenüber ausgehenden Internetzugriffen
- Prävention der Überbeanpruchung von Rechenressourcen
- Logging auf der Netzwerkebene

PePP

Jupyterhub – Vorteile

- Keine lokalen Ressourcen und Installationen nötig
- Niederschwelliger Zugang (Software as a Service)
- Vordefinierte Umgebungen
  - Auswahl an Programmiersprachen und Anwendungen
  - Modifizerbarkeit durch zusätzliche Rollen zwischen Benutzer und Administrator
  - Wartung durch den Serviceanbieter
- Kollaboratives Arbeiten
- Bereitstellung spezialisierter Hardware (z.B. GPUs, TPUs, etc.)
- Systemübergreifender Austausch (z.B. Anbindung an das LMS oder Github)

PePP

Repositories und Rollout

- Jupyter ILIAS-Plugin für Tests (Release 8)
  - https://github.com/TIK-NFL/jupyter-ilias-plugin
- Jupyterhub für integrierte Benutzung
  - https://github.com/TIK-NFL/jupyterhub

- ViPLab ILIAS-Plugin für Tests (Release 8, auslaufend)
  - https://github.com/TIK-NFL/ViPLab

- Rollout
  - Hochschuleigene Infrastruktur für Jupyterhub notwendig (Docker-Betrieb)
  - Untersützung durch das Reallabor (RL3.1) zur Projektlaufzeit möglich

PePP

Weiterführende Projekte und Ausblick

- bwJupyter
  - Landesweite Verwendung an Hochschulen in der Lehre als niederschwellige LMS-Integration
  - OER-Austausch und Wiederverwendbarkeit
  - Mögliche Weiterentwicklung und Verteilung im Kontext der Lehre
- bwCloud3
  - Aufbau eines leistungsfähigen und landesweiten Kubernetes-Clusters
  - Provisionierung spezieller Rechenressourcen, z.B. GPU
- NFDI4Ing
  - Expertise im Bereich Forschungsdatenmanagement → Reproduzierbarkeit
  - Spezialisierte Jupyter-Images für Ingenieurwissenschaften

PePP
Partnerschaft für innovative E-Prüfungen
Projektverbund der baden-württembergischen Universitäten

Praxisdemo: „Prüfungsworkflow“

»Partnerschaft für innovative E-Prüfungen. Projektverbund der baden-württembergischen Universitäten (PePP)«

gefördert von der »Stiftung Innovation in der Hochschullehre«

PePP
Partnerschaft für innovative E-Prüfungen
Projektverbund der baden-württembergischen Universitäten

ILIAS® ViP Lab jupyter

Interaktiver Teil

https://vp.tik.uni-stuttgart.de/

»Partnerschaft für innovative E-Prüfungen. Projektverbund der baden-württembergischen Universitäten (PePP)«

universität freiburg · Universität Heidelberg · Universität Hohenheim · KIT Karlsruher Institut für Technologie · Universität Konstanz · Universität Mannheim · Universität Stuttgart · Eberhard Karls Universität Tübingen · universität ulm

gefördert von der »Stiftung Innovation in der Hochschullehre«

Stiftung Innovation in der Hochschullehre

PePP
Partnerschaft für innovative E-Prüfungen
Projektverbund der baden-württembergischen Universitäten

Vielen Dank!

Admir Obralija, M.Sc. - admir.obralija@tik.uni-stuttgart.de
Ina Baret - ina.baret@zlw.uni-stuttgart.de
Oliver König, M.Sc. - oliver.koenig@ians.uni-stuttgart.de

Universität Stuttgart
Technische Informations- und Kommunikationsdienste (TIK)
Allmandring 30
70569 Stuttgart

Telefon: ++49-711-685-63833
Web: www.tik.uni-stuttgart.de

PePP-Gesamtkoordination
Elisa Bumann
Universität Freiburg
Rechenzentrum
Elisa.bumann@rz.uni-freiburg.de
www.hnd-bw.de/pepp

»Partnerschaft für innovative E-Prüfungen. Projektverbund der baden-württembergischen Universitäten (PePP)«

gefördert von der »Stiftung Innovation in der Hochschullehre«