

George Dogaru (Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen), Ralf Klammer (Technische Universität Dresden)

## Infrastruktur

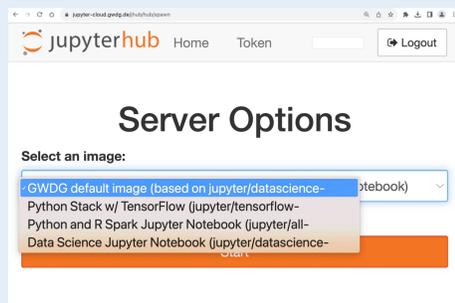
Notebooks können auf der eigenen Maschine oder in der Cloud ausgeführt werden. Letzteres ist für Text+ über den JupyterHub unter [jupyter-cloud.gwdg.de](https://jupyter-cloud.gwdg.de) möglich.

Der Dienst ist an die Single Sign-on Infrastruktur der GWGDG angebunden, was die Nutzung vieler Dienste über eine einzige Anmeldung ermöglicht.

Eine aufwändige Installation oder Aktualisierung von Jupyter entfällt bei der Nutzung der Cloud. Diese ermöglicht die sofortige Interaktion mit der Plattform und stellt einen erheblichen Vorteil für Einsteiger:innen dar. Darüber hinaus werden in der Task Area Infrastructure/Operations (IO) entwickelte Lösungen auf ihre Lauffähigkeit in der GWGDG Jupyter Cloud geprüft.

Ein Hindernis bleibt dennoch: Verfügbare Notebooks müssen zunächst geholt (heruntergeladen, hochgeladen oder mit git geklont) und eventuelle Abhängigkeiten installiert werden.

Es gibt deswegen mehrere Ansätze und Bestrebungen, den Installations- und Konfigurationsaufwand zu eliminieren. Die GWGDG Jupyter Cloud bietet bereits die Möglichkeit, zwischen mehreren vor-konfigurierten Umgebungen (Images) zu wählen (siehe Abbildung).



Die Erstellung eines Images, welches die in IO produzierten Notebooks enthält, wäre eine Möglichkeit, die Notebooks sofort einsatzbereit zu machen.

Außerdem ist die Nutzung von mehr Storage sowie von High Performance Computing (HPC) Ressourcen für Text+ geplant.

## Beispiel 2: UI für Tool zur Vereinfachung des Datenimports in TextGrid

Für das Einspielen von Daten in das TextGrid-Repository – ohne TextGridLab, welches immer seltener Verwendung findet – werden TextGrid-spezifische Metadaten-Dateien benötigt. Die Erstellung dieser Dateien kann weitgehend automatisiert werden.

Dafür entsteht in IO/M5 ein Tool, das die Metadaten aus den einzuspielenden Dateien liest, deren Korrektur/Ergänzung erlaubt, und schließlich die benötigten Dateien erstellt.

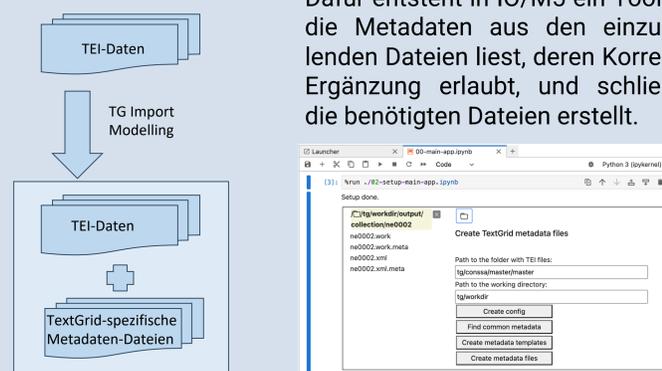
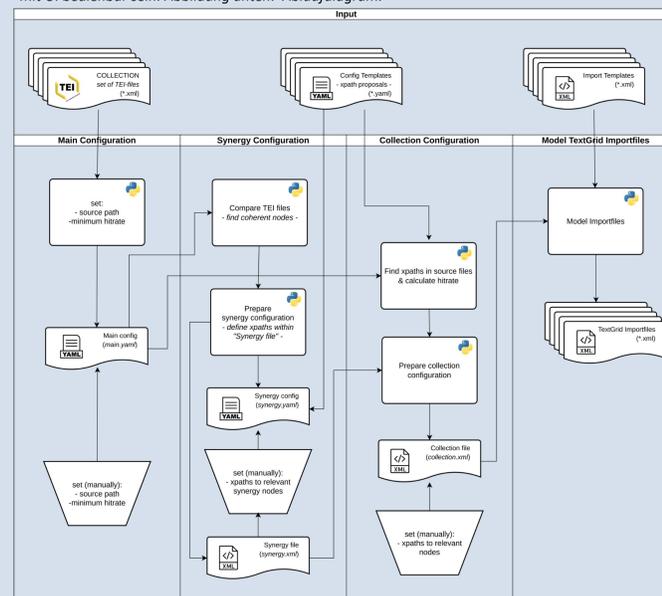
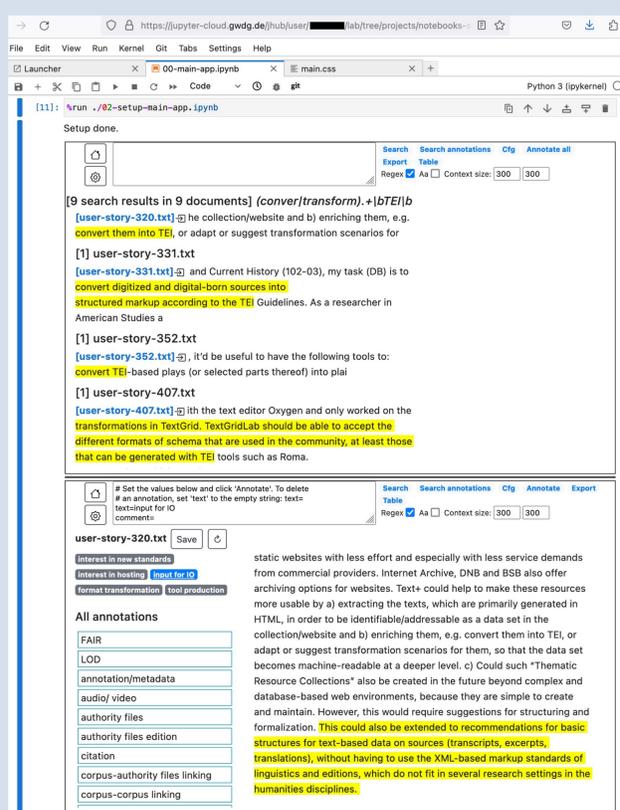


Abbildung oben rechts: das Kommandozeilen-Tool wird auch über eine Notebook-basierte Anwendung mit UI bedienbar sein. Ablaufdiagramm.



## Beispiel 1: Plaintext-Dateien durchsuchen und annotieren



Ein Notebook-basiertes Tool zum Suchen und Annotieren von Plaintext-Dateien (hier der Text+ User Stories).

Die zwei Output-Zellen sind interaktiv. Sie sind mit üblichen Webtechnologien (HTML, CSS & JavaScript) realisiert. Jupyter Widgets ermöglichen hierbei eine Aktualisierung der Zellen, um diese mit neuen Inhalten zu befüllen. Die Kommunikation zwischen Python und JavaScript erfolgt über Dateien. Der Code für das UI lässt sich wiederverwenden, so dass mit wenig Aufwand weitere interaktive Notebook-basierte Tools erstellt werden können.

## Angebote, Perspektiven

Mehrere Python-basierte Notebooks existieren bereits. Die zuerst entstandenen Notebooks sind *Demos* [1], die verdeutlichen, wie verschiedene Aufgaben sich mit Python bzw. mit Notebooks lösen lassen. Beispielsweise:

- XML-Datenverarbeitung mit XSLT 3
- Datengewinnung aus HTML mittels BeautifulSoup
- Verarbeitung von tabellarischen Daten mittels pandas

Ein interaktives Tool zum Durchsuchen und Annotieren von Plaintext-Dateien [2] ist ebenso verfügbar. Das Tool ist ähnlich einer Web-basierten Anwendung bedienbar und das User Interface (UI) wird mit den typischen Web-Technologien (HTML, CSS, JavaScript) umgesetzt.

Die Code-Basis zur Erstellung von UIs innerhalb von Notebooks lässt sich wiederverwenden und wird aktuell für ein Python-Tool zur Vereinfachung des Datenimports in TextGrid [3] nachgenutzt.

Ein breites Spektrum an Digital Humanities-Aufgaben lassen sich mit Notebooks lösen. Dazu zählen unter anderem Formatkonvertierungen, Datenanreicherungen oder Aufbereitung von Daten für Anzeige oder Import in Repositorien. Bei der Planung weiterer Tools im Kontext der IO-Aufgaben „Data Processing Pipelines“ und „Software Services“ spielt Input aus der Community eine wichtige Rolle. Bereits geplant sind darum:

- ein Tool zur Konvertierung von Text-Dateien in TEI-XML
- ein Tool zur Ausführung von XSLT-Transformationen

Die Bedienung soll jeweils über Notebooks funktionieren.

- [1] [https://gitlab.gwdg.de/textplus/jupyter-demos-2022/tree/v0.1.0?ref\\_type=tags](https://gitlab.gwdg.de/textplus/jupyter-demos-2022/tree/v0.1.0?ref_type=tags)  
 [2] [https://gitlab.gwdg.de/textplus/notebooks-search-annotate/-/tree/v0.1.1?ref\\_type=tags](https://gitlab.gwdg.de/textplus/notebooks-search-annotate/-/tree/v0.1.1?ref_type=tags)  
 [3] [https://gitlab.gwdg.de/textplus/textplus-io/textgrid\\_import\\_modelling/-/tree/v0.1.4?ref\\_type=tags](https://gitlab.gwdg.de/textplus/textplus-io/textgrid_import_modelling/-/tree/v0.1.4?ref_type=tags)