

Architektur- Überblick

Diese Übersicht beschreibt den Stand der D-DOK-Komponenten (Release 0.0.7, in Entwicklung 0.0.8). Quelle: git.agrarforschung.at/ddok/ddok.

Architektur-Diagramm

```
flowchart LR
  subgraph Client["Browser-Client"]
    NG["Angular 21 SPA  
(client/)"]
  end

  subgraph Server["D-DOK Server (Node.js)"]
    API["Express + Inversify  
(server/)"]
    PDF["PDF-Parse Worker  
(modules/pdfparse)"]
  end

  subgraph DBs["Datenbanken (lokal)"]
    MY["MySQL  
Projekte / Publikationen  
Metadaten"]
    MO["MongoDB 7.0  
PDF-Dateien + Volltext"]
  end

  subgraph DESS["D-ESS"]
    DESSDB["MySQL  
Kostentraeger / Benutzer:innen"]
  end

  subgraph Auth["Authentifizierung"]
    AD["Active Directory  
(LDAP)"]
    KC["Keycloak OIDC  
sso.agrarforschung.at"]
  end

  subgraph Web["BAB Website (0.0.8)"]
    WEBMY["MySQL  
WEB_DATABASE"]
    WEBMO["MongoDB  
WEB_MONGO"]
    SITE["bab.gv.at Public-Frontend"]
  end

  NG -- HTTPS --> API
  API --> MY
  API --> MO
  API -- Lookup --> DESSDB
  API -. optional .-> AD
  API -. optional .-> KC
  API -- "Sync (0.0.8)" --> WEBMY
  API -- "Sync (0.0.8)" --> WEBMO
```

```
WEBMY --> SITE
WEBMO --> SITE
PDF --> MO
```

```
classDef new fill:#e8f5e9,stroke:#388e3c
class WEBMY,WEBMO,SITE,Web new
```

Komponenten im Überblick

Komponente	Stack	Quelle
Frontend (SPA)	Angular 21 (seit 0.0.8, vorher Angular 17) · Wijmo · Quill-Editor	<code>client/</code>
Backend-API	Node.js · Express · Inversify (DI) · TypeScript	<code>server/</code>
PDF-Parser	Worker für Volltext-Extraktion aus PDF-Dateien	<code>modules/pdfparse/</code>
Relationale DB	MySQL 3306 — Projekte, Publikationen, Stammdaten	D-DOK eigene DB
Dokumenten-DB	MongoDB 7.0 (Port 27017) — PDF-Binaries + Volltext-Index	D-DOK eigene DB
D-ESS-Anbindung	Read-only Zugriff auf D-ESS-Datenbank für Kostentraeger/KS/KT und Benutzer:innen-Stammdaten	<code>DESS_LINK_ENABLED=true</code>
Authentifizierung	Lokal · Active Directory (LDAPS) · Keycloak OIDC (Realm <code>ddok</code> auf <code>sso.agrarforschung.at</code>)	Konfiguration in <code>server/config/config.js</code>
Website-Sync (NEU 0.0.8)	Zusätzliche MySQL- und MongoDB-Verbindung zur öffentlichen BAB-Website — Export ausgewählter Publikationen	<code>WEB_DATABASE_*</code> + <code>WEB_MONGO_DATABASE_*</code>

Deployment

- **Server:** Dedizierter Anwendungsserver (Debian) mit Node.js, lokaler MySQL- und MongoDB-Instanz.
- **SSO-Integration:** Keycloak-Realm `ddok` (vgl. Eintrag in der *SSO-Anwendungsliste* auf `sso.agrarforschung.at`).
- **Build:** `npm run build` — baut Client, Server (via `esbuild`) und PDF-Parser-Modul; Docker-Image via `Dockerfile` im Repo-Root.
- **CI/CD:** `.gitlab-ci.yml` im D-DOK-Repository.

Datenfluesse

1. **Projekt-/Publikationsanlage:** Benutzer:innen erstellen Projekte und Publikationen im Angular-Frontend; Daten werden in der MySQL-DB persistiert.
2. **PDF-Upload:** Hochgeladene Dateien werden in MongoDB gespeichert; ein PDF-Parser-Worker extrahiert den Volltext für die Suche.
3. **D-ESS-Lookup:** Kostenstellen, Kostentraeger und Benutzer:innen-Stammdaten werden zur Laufzeit aus der D-ESS-Datenbank gelesen.
4. **Website-Export (0.0.8):** Freigegebene Publikationen werden in die separate Website-Datenbank synchronisiert und auf der öffentlichen BAB-Website angezeigt.

Letzte Aktualisierung: 2026-05-23 · Pflege: Roland Neissl · Quelle: git.agrarforschung.at/ddok/ddok

🕒 Version #3

★ Erstellt: 2026-05-23 15:39:11 CEST von roland.neissl

🔧 Zuletzt aktualisiert: 2026-05-23 17:33:59 CEST von roland.neissl