

VuFind PAIA-ILS-Treiber

Vorstellung des PAIA ILS-Treibers für VuFind

UNIVERSITÄTS
BIBLIOTHEK *ubl* LEIPZIG



VuFind PAIA-ILS-Treiber

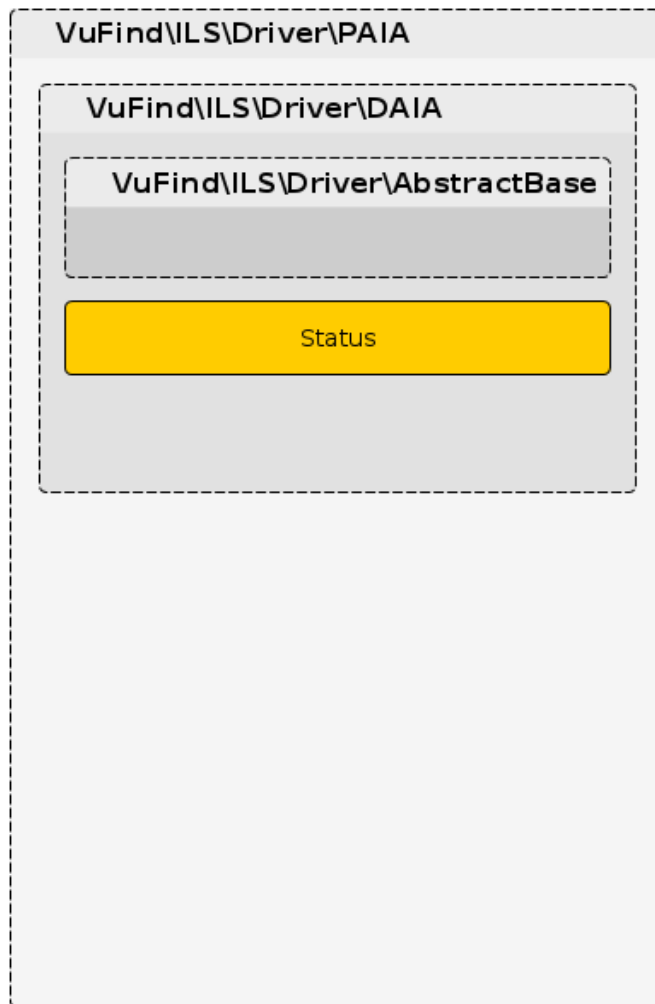
- *Basiert auf GBV PAIA Treiber*
- *Weiterentwicklung durch Universitätsbibliothek Leipzig und tub Hamburg*
- *Seit VuFind 3.1 verfügbar*



VuFind PAIA-ILS-Treiber

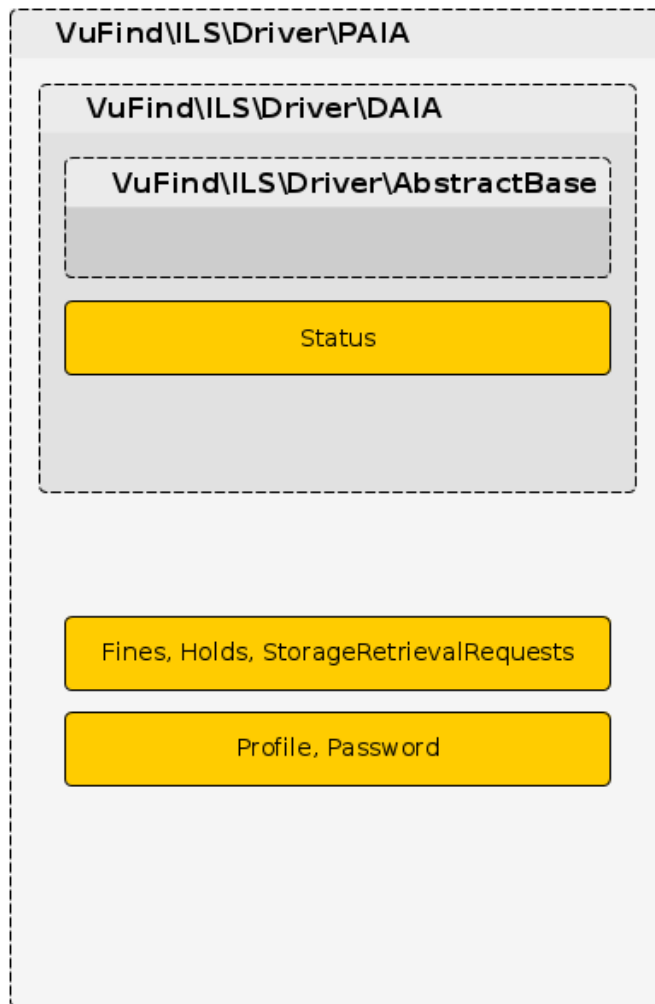
- *Überblick über den Treiber:
DAIA/PAIA-Stack · Funktionsumfang · Konfiguration ·
Caching · Anpassungsmöglichkeiten*
- *PAIA Einsatz mit VuFind – Erfahrungen und Ausblick*

DAIA/PAIA Stack



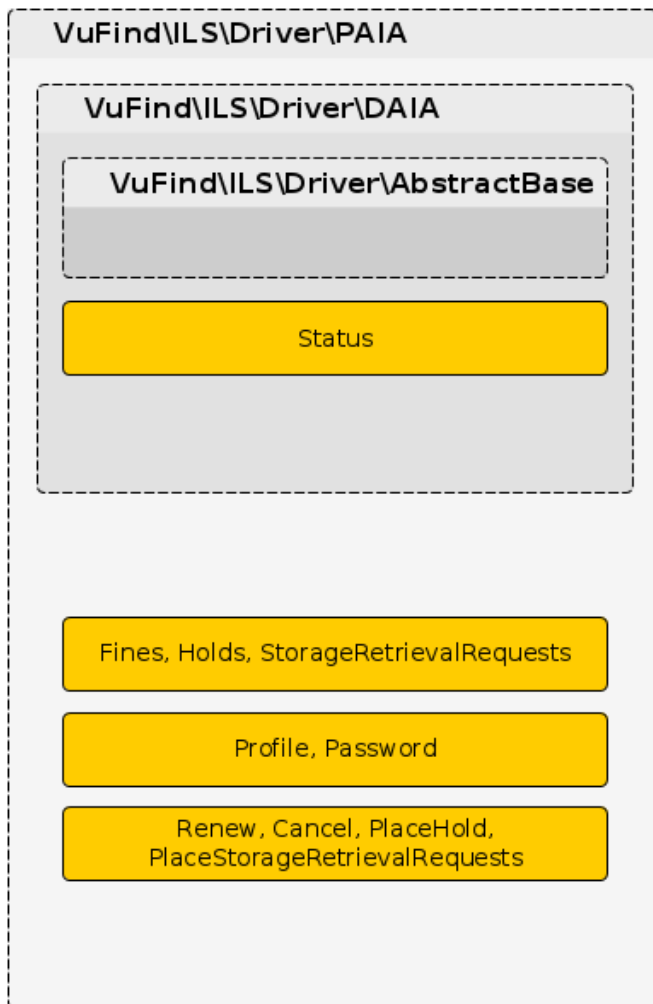
- *PAIA erweitert den DAIA-Treiber über Vererbung*
 - DAIA für Verfügbarkeitsinformationen
 - PAIA für Nutzerkontofunktionen

PAIA - Kontofunktionen



- *Profilfunktionen (Profildaten, Passwortänderung)*
- *Item-Views (Ausleihen, Magazinbestellungen, Vormerkungen)*
- *Gebühren*

PAIA - Interaktionen mit dem Katalog



- *Bestellen (Magazinbestellung) und Vormerken*
- *Verlängerung, Abbruch von Magazinbestellungen, Vormerkungen*

PAIA - Konfiguration

- *PAIA.ini*:

```
[PAIA]
baseUrl = "" ; base URL des PAIA Servers

; optional
timeout = 30 ; Timeout in Sekunden für PAIA HTTP-Requests
paiaCache = false ; Caching in PAIA

[Holds] ; Konfiguration für Vormerkungen
HMACKeys = "id:item_id:doc_id"

[StorageRetrievalRequests] ; Konfiguration für Magazinbestellungen
HMACKeys = "id:item_id:doc_id"

[Renewals] ; Konfiguration für Verlängerungen
HMACKeys = "id:item_id:doc_id"
```

PAIA – Caching

- ***DAIA/PAIA teilen sich Caching***
 - Standardmäßig deaktiviert
 - Cache-TTL für DAIA und PAIA identisch (90s)
- ***DAIA – Caching:***
 - Auf Record-Ebene (ein Cache-Objekt pro Record)
 - Nach Bestellvorgänge etc. wird DAIA Cache für Record aktualisiert
- ***PAIA – Caching:***
 - Auf Nutzerebene (ein Cache-Objekt pro Nutzer)
 - Alle Records pro User → bei Verlängerung etc. wird der Cache aktualisiert

PAIA - Mapping VuFind-Methoden

- *PAIA liefert immer alle mit dem Nutzer verknüpften Items*
- *VuFind bietet unterschiedliche Views für Statuskategorien (bspw. CheckedOut, Holds, StorageRetrievalRequests, etc.)*
- *Mapping notwendig:*
 - 1. PAIA-Status → Views
 - 2. PAIA-Datenmodell → VuFind View-Datenmodell

PAIA - Mapping leicht gemacht I

- *Anpassung für institutionsspezifische Views unproblematisch umzusetzen*
- *Permanentausleihen?*
 - Entscheidendes Kriterium: kein Rückgabedatum
- *Lesesaalexemplare?*
 - Entscheidendes Kriterium: document.storageid
- *Fernleihen?*
 - Mögliches Kriterium: URI mit fremden Namespace

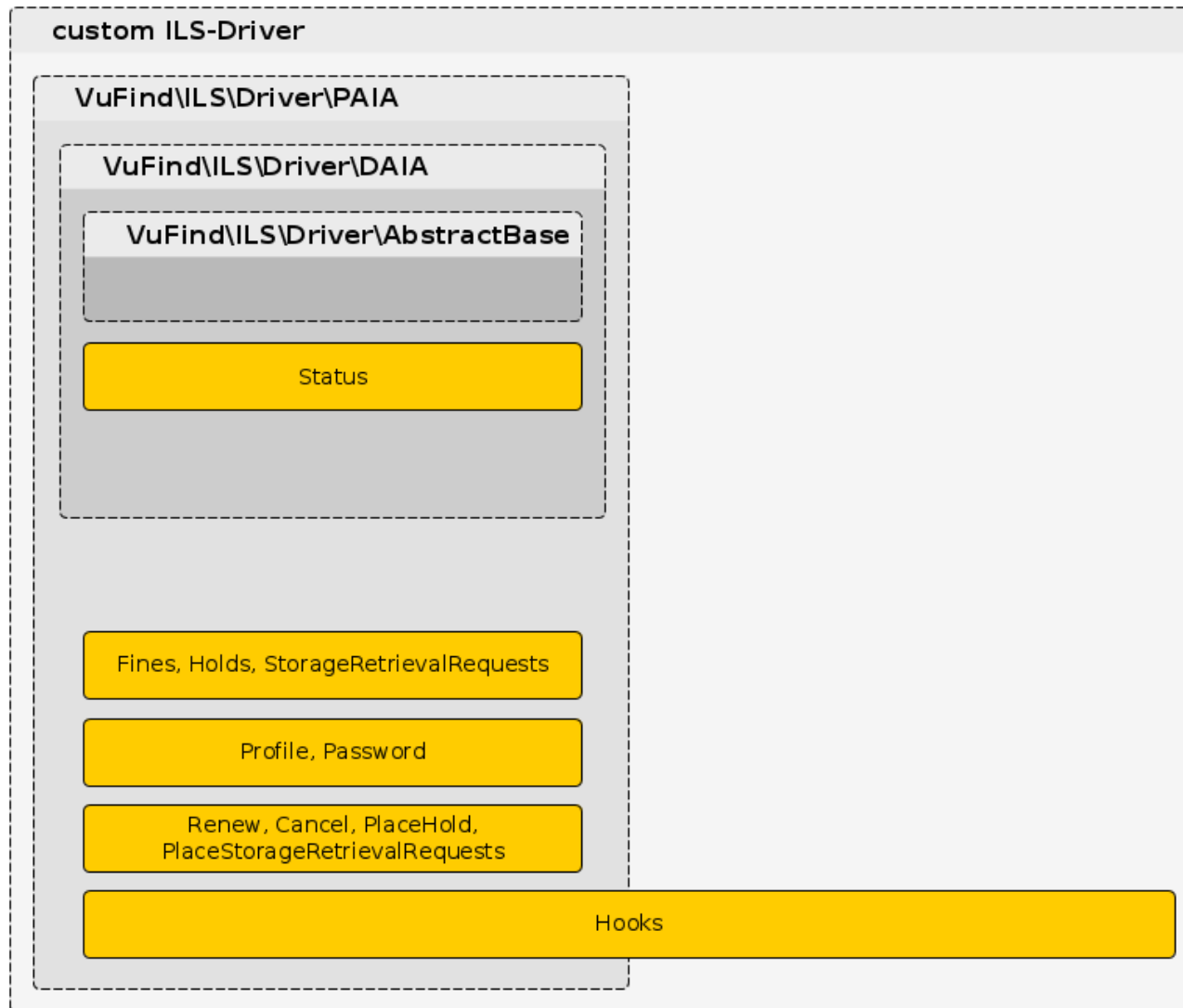
PAIA - Mapping leicht gemacht II

- z.B. *getMyHolds()* (alle Exemplare, die entweder vorgemerkt oder bereit zum Abholen sind):

```
public function getMyHolds($patron)
{
    // filters for getMyHolds are:
    // status = 1 - reserved (the document is not accessible for the patron yet,
    //     but it will be)
    //     4 - provided (the document is ready to be used by the patron)
    $filter = ['status' => [1, 4]];
    // get items-docs for given filters
    $items = $this->paiaGetItems($patron, $filter);

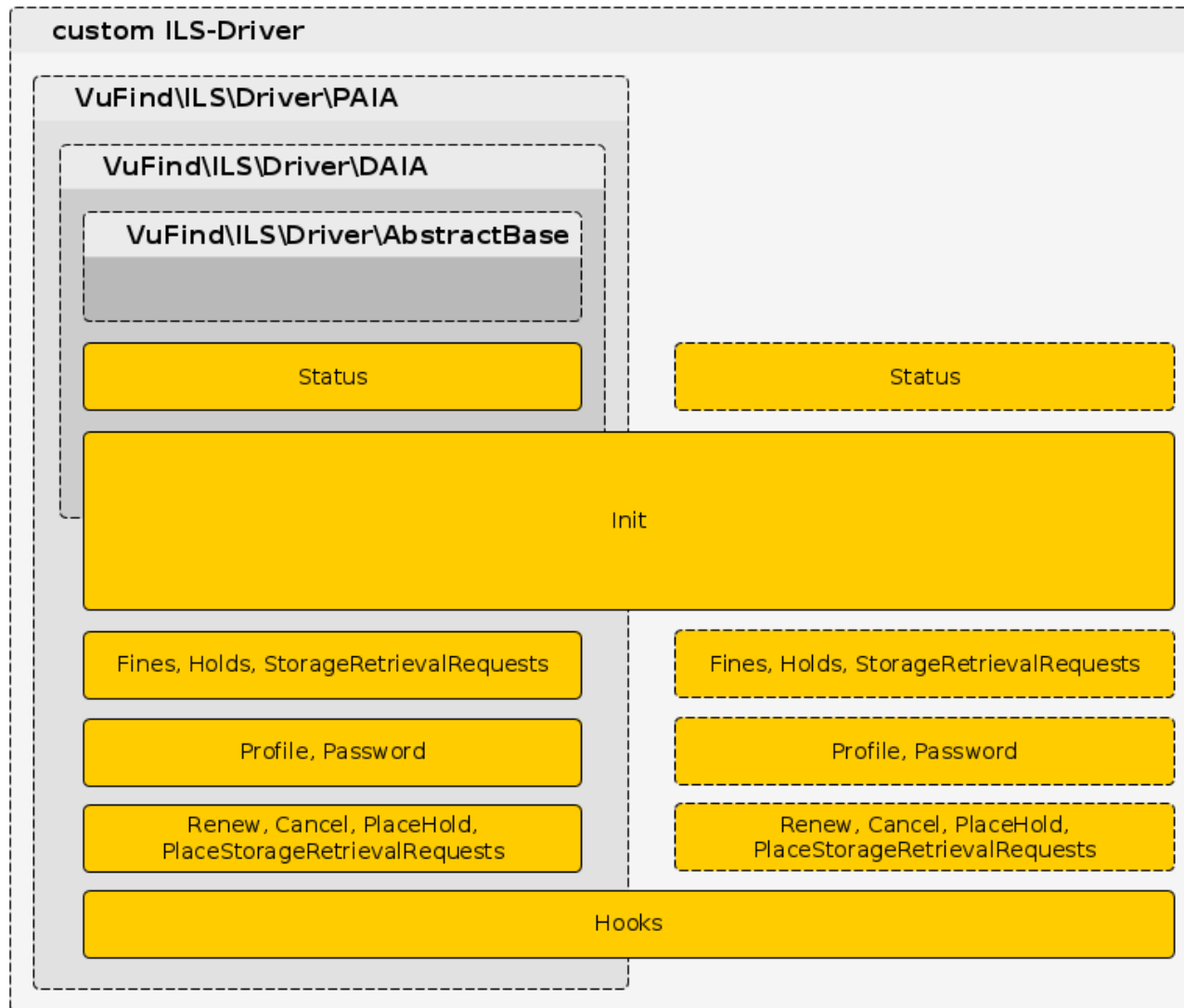
    return $this->mapPaiaItems($items, 'myHoldsMapping');
}
```

PAIA - Hooks



- *Institutions- und Katalogspezifischen Anpassung in Custom-Driver*
- *sowohl in DAIA als auch in PAIA*
- *relevant in PAIA*
getAlternativItemId()
 - Übersetzung von VuFind-Id nach PAIA-Id

PAIA - Individuelle Anpassungen



PAIA bei **finc**

- *finc-Nutzergemeinschaft mit 15 Mitgliedern*
- *Heterogene ILS-Landschaft*
- *Vereinheitlichung der ILS-Anbindung*
- *Standardisierte API für Verfügbarkeitsinformationen*
- *Fertige Java DAIA/PAIA-Services für Libero 5.5, Libero 6 und Bibliotheca*



• PAIA Einsatz mit VuFind - Erfahrungen und Ausblick

- *PAIA-Ausblick:*

- Profildaten-Änderung (In Arbeit)
- Performance-Optimierung (Items liefert alle Exemplare)
- Ausbau OAuth2 Authentifizierungs-Methoden
- Systemnachrichten vom ILS an den Nutzer

- *VuFind-Ausblick:*

- Aushandeln von Bedingungen bei ILS-Interaktionen

VuFind PAIA-ILS-Treiber

Vielen Dank!

lahmann@ub.uni-leipzig.de