FOLIO

EINFÜHRUNG

Richard Redweik
redweik@ub.uni-leipzig.de
Universitätsbibliothek Leipzig

28.09.2017

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

AGENDA

1. Motivation
2. FOLIO
   a. Übersicht
   b. Okapi
   c. Stripes
3. Zusammenfassung

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

MOTIVATION

Warum FOLIO?

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

MOTIVATION

- Neuimplementierung von amsl
- Electronic Resource Management System für Bibliotheken auf Basis von Linked Data Technologien
- Kann Neuimplementierung auf FOLIO aufsetzen?

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

FOLIO

the *future* of libraries is
OPEN

UNIVERSITÄTS
BIBLIOTHEK ubl LEIPZIG

FOLIO?

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

ALLGEMEINE ÜBERSICHT

- Open Source Plattform
  - Entwicklung einer technischen Plattform: Library Service Platform (LSP)
  - Darauf aufbauend: Bibliotheksmanagementsystem
- Offen für Bibliotheken & komm. Anbieter
  - Nutzung
  - Entwicklung

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

COMMUNITY

- OLE Community (Open Library Enviroment)
  - Cornell University Libraries, Texas A&M University Libraries, Duke University Libraries, GBV (Göttingen), hbz (Köln), University of Chicago Library, Lehigh University, North Carolina State University Libraries, SOAS (London), Andrew W. Mellon Foundation
- EBSCO (haupts. Finanzierung)
- Index Data (Entwicklung)

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

ROADMAP

- **2017**:
  - Initiale Apps (Circulation, Rechtemanagement, Nutzermanagement...), UI Design, Dokumentation, User Guides
- **2018**:
  - Erste Bibliothek live (Initiale Version)
  - Vorbereitung 5 weiterer Bibliotheken
  - Selbsttragende Open-Source Community
  - Service und Hosting Optionen von Anbietern

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

TECHNISCHER ÜBERBLICK

- LSP ist Grundlage für funktionale Module (z.B. Ausleihe, amsl...)
- Basiert auf Microservice-Idee: "Plug-and-Play" Infrastruktur
  - Unabhängige Entwicklung
  - Modular: Einzelne Nutzung und Installation
- Einheitliche Benutzeroberfläche

→ Module flexibel erweiter- und austauschbar

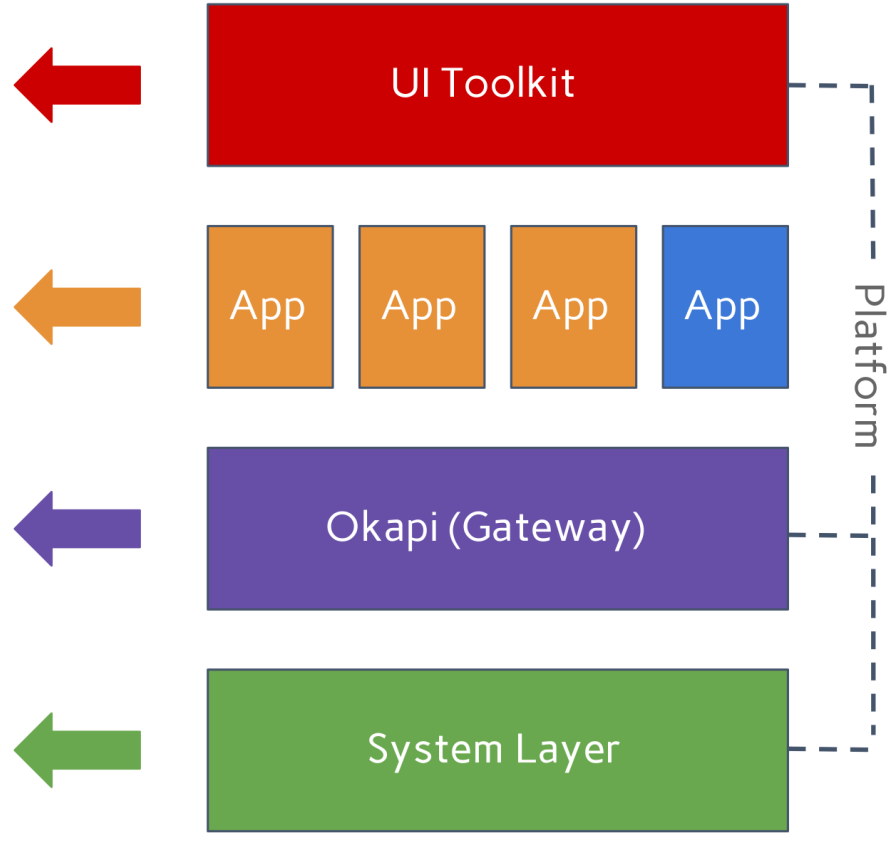
UNIVERSITÄTS BIBLIOTHEK LEIPZIG

Einheitliche Benutzeroberfläche. Kann genutzt oder adaptiert werden.

Basis Folio-Apps (ERM, Ausleihe, Katalogisierung...)
Eigene Apps bzw. von Drittanbietern

Zentraler Kommunikationsknoten
Regelt Kommunikation und Trennung zwischen Apps und Mandanten

Zentrale Datenschicht (DB)
Indexing, Logging, Mandantenkonfiguration



UNIVERSITÄTS BIBLIOTHEK LEIPZIG

OKAPI

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

OKAPI - ÜBERSICHT

- FOLIO Middleware
- Agiert als API Gateway
  - Alleiniger Einstiegspunkt ins System
- Layer zum Management von FOLIO Apps und Services
- Hauptaufgaben: Deployment, Monitoring, Service Discovery, Caching...

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

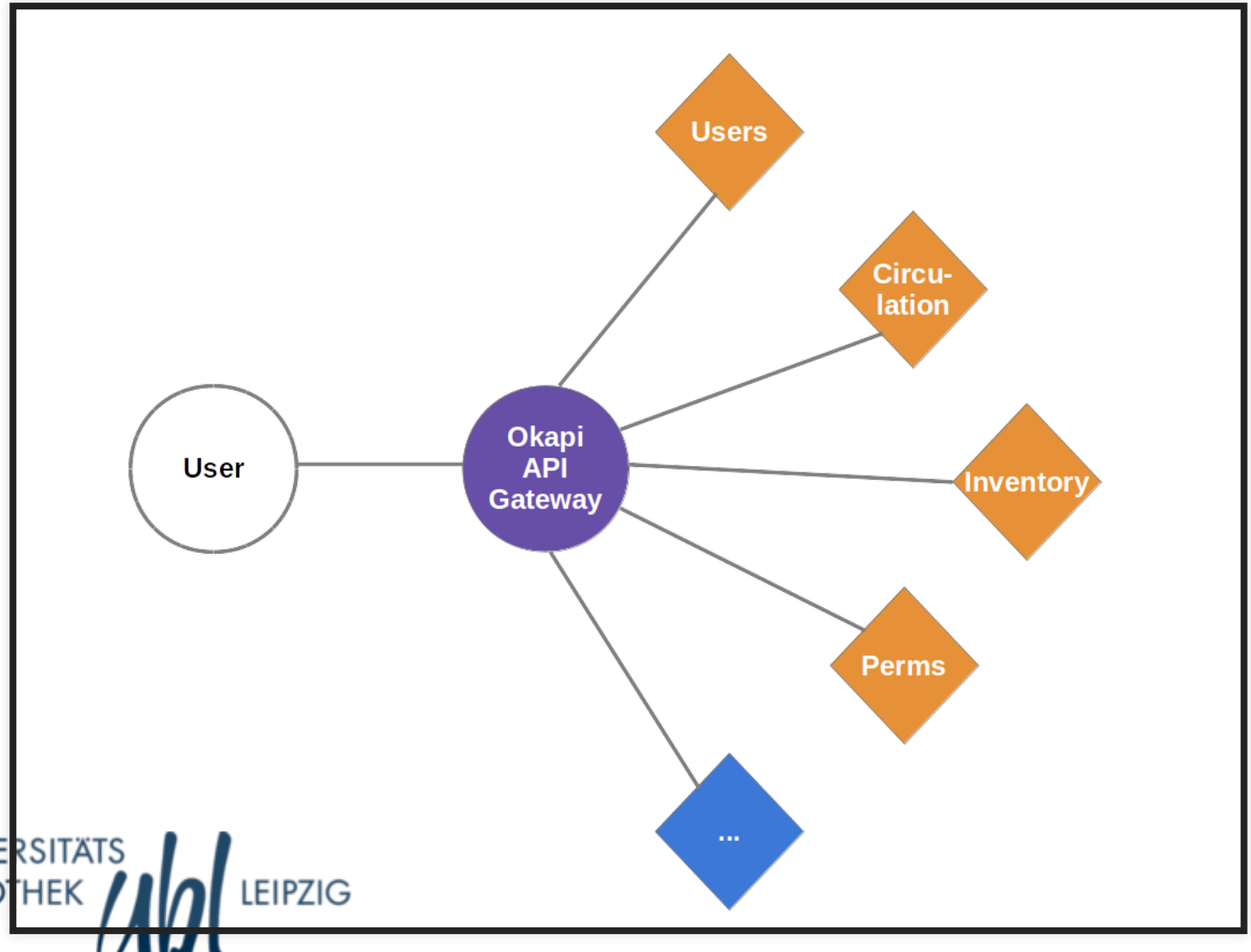
API GATEWAY PATTERN

Okapi implementiert das *API Gateway* Pattern:

> *“the API Gateway encapsulates the internal system architecture and provides a unified API that may be tailored to each client; it might also include core responsibilities such as authentication, monitoring, load balancing, caching, request shaping and management, and static response handling.”*

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

API GATEWAY PATTERN



UNIVERSITÄTS BIBLIOTHEK LEIPZIG

OKAPI: WEITERE AUFGABEN

- Load Balancing
- Isolation von Mandanten
- Prüfung von Nutzerrechten

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

OKAPI

- Konfigurier- und erweiterbar
- Neue oder existierende Web Service Endpunkte bzw. Module können ohne programmatische Änderungen hinzugefügt werden
- Wichtigste Voraussetzung:
  RESTful HTTP Netzwerkkommunikation

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

OKAPI - ARCHITEKTUR

- Okapi macht wenige Annahmen über API Endpunkte einzelner Module
- Routing von HTTP Operationen (POST, GET, PUT, DELETE) zu registrierten Endpunkten

→ Plattform basiert auf RESTful HTTP Interface Contracts

→ Module unabhängig von Programmiersprache

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

OKAPI'S WEB SERVICES

- Basisfunktionalitäten zum
  - Aufsetzen, Konfigurieren und Aktivieren von Modulen
  - Management von Mandanten

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

OKAPI MODULE

- Module müssen folgende Anforderungen erfüllen
  - RESTful HTTP Netzwerkkommunikation
  - **ModuleDescriptor.json**: Beschreibt Metadaten, Abhängigkeiten zu dritten Modulen und alle Interfaces
  - **DeploymentDescriptor.json**: Enthält Liste der Modul-Routen: Ermöglicht Okapi Anfragen an Modul weiterzuleiten (Proxy)

→ Module werden durch ihr Verhalten und nicht ihren Inhalt definiert!

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

STRIPES

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

STRIPES: ÜBERSICHT

- User Interface (UI) Toolkit zur Erstellung von FOLIO Web-Applikationen
- Ziel: Einfache Erstellung von UI Modulen, die mit RESTful Webservices kommunizieren
- Besteht aus mehreren JS Bibliotheken: stripes-connect, -components, -loader, -logger, -core

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

STRIPES: TECHNISCHE GRUNDLAGE

- Stripes UI Module geschrieben in
  - **JavaScript** (ES6) plus
  - **React**
- Damit kommt ganzes JS-Ökosystem: Babel, Webpack, Yarn

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

REACT

- Bibliothek zu Erstellung von User-Interface-Komponenten von Facebook
- "[...] will efficiently update and render just the right components when your data changes."
- Komponenten kommunizieren mit Okapi
- Ziel: Einfacheres Schreiben von Code, dessen Bestandteile weniger verschränkt und verwoben sind

ZURÜCK ZU STRIPES

Stripes besteht aus einzelnen Komponenten:

- **stripes-connect**: Managed Verbindung zu FOLIO/Okapi Services
- **stripes-components**: Bietet nutzbare UI-Komponenten (React) an (Checkboxen, Suchfelder, Multi-Panes...)
- **stripes-loader**: Pulled mehrere Stripes-Module in eine Web-Applikation
- **stripes-logger**: Logging
- **stripes-core**: Web-Applikation, die UI-Module kontrolliert

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

STRIPES KOMPONENTEN

- Erleichtern Erstellung von Web-Applikationen
  - *stripes-connect*: Einfache Nutzung von Okapi-Modulen via Rest
  - *stripes-components*: Schnelle Erstellung einer React Anwendung durch Nutzung von UI-Komponenten (z.B. MultiColumnList, Pane)

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

ZUSAMMENFASSUNG

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

TECHNISCH

- FOLIO bietet nutzbares Ökosystem an
  - Enthält nutzbare Module
  - LSP kann als Basis für neue Module genutzt werden
  - Klare Trennung zwischen Frontend and Backend. Beides kann unabhängig ausgetauscht werden.

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

OKAPI

- Nutzt Microservice Pattern
- Stellt API Gateway Implementierung bereit
- Macht Registrierung, Deployment und Management von Modulen einfach

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

STRIPES

- Toolkit zum Erstellen von UI Modulen
- Bietet wiederverwendbare Komponenten und Funktionalitäten an
- Ermöglicht schnelles Erstellen von React-Anwendungen

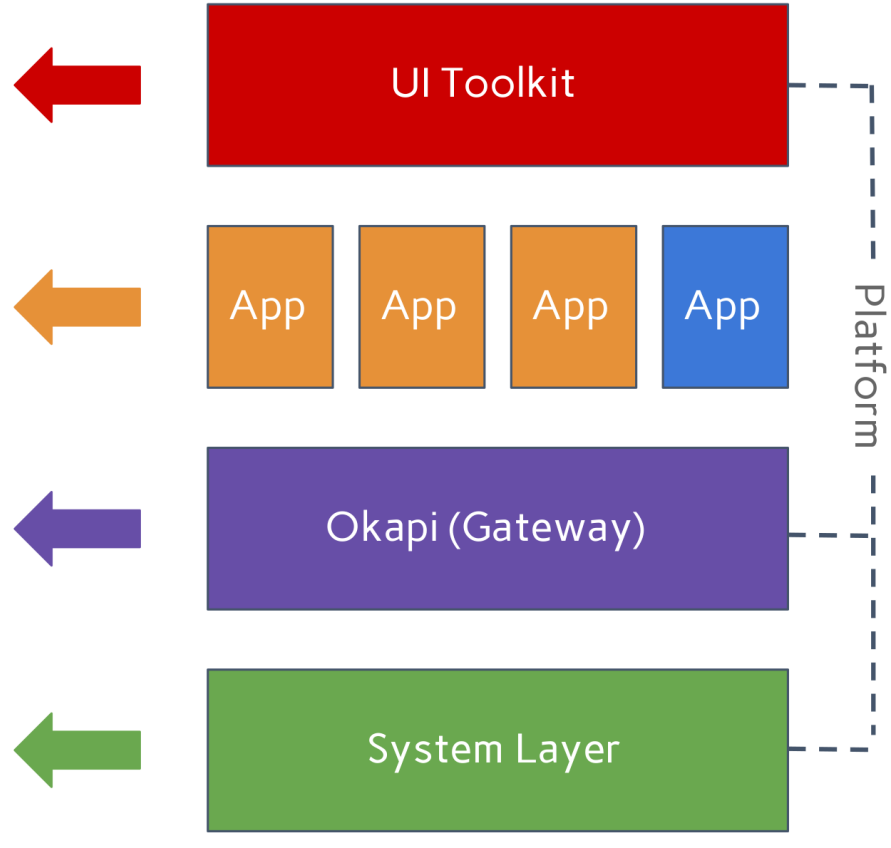
UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG

Einheitliche Benutzeroberfläche. Kann genutzt oder adaptiert werden.

Basis Folio-Apps (ERM, Ausleihe, Katalogisierung...)
Eigene Apps bzw. von Drittanbietern

Zentraler Kommunikationsknoten
Regelt Kommunikation und Trennung zwischen Apps und Mandanten

Zentrale Datenschicht (DB)
Indexing, Logging, Mandantenkonfiguration



UNIVERSITÄTS BIBLIOTHEK LEIPZIG

FRAGEN?

Richard Redweik
redweik@ub.uni-leipzig.de
Universitätsbibliothek Leipzig

UNIVERSITÄTS BIBLIOTHEK LEIPZIG

QUELLEN / LINKS

- http://dev.folio.org/doc/glossary.html
- https://github.com/folio-org/okapi/blob/master/doc/guide.md
- https://github.com/folio-org/stripes-core/blob/master/doc/dev-guide.md
- http://www.indexdata.com/folio/repositories-whitepaper
- https://www.nginx.com/blog/building-microservices-using-an-api-gateway/
- https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/2938
- https://wiki.folio.org/display/PC/FOLIO+Roadmap

UNIVERSITÄTS BIBLIOTHEK ubl LEIPZIG