

Das  
Bibliotheksmanagementsystem  
FOLIO

—  
Aktueller Stand von Plattform und  
Funktionalität

Maike Osters, hbz

Julian Ladisch, VZG

107. Bibliothekartag, Berlin

12. Juni 2018

FOLIO

—

Bibliotheksmanagementsystem





# FOLIO im Überblick

- Historie

- Das Open-Source-Projekt FOLIO wurde von den Stakeholdern EBSCO, Index Data und der OLE-Community angestoßen
- In Entwicklung seit 2016
- Im selben Jahr Bildung einer neuen Dachorganisation: Open Library Foundation

- Ziel

- Konzeption und Entwicklung einer Library Service Platform (LSP) und der Aufbau eines neuen Bibliotheksmanagementsystems (BMS) „FOLIO“ auf dieser Plattform

➔ Offen, nachhaltig, innovativ, flexibel, erweiterbar



# FOLIO im Überblick

- Modernes UX/UI-basiertes Entwicklungskonzept
  - Anpassbar auf individuelle Bedürfnisse von Bibliotheken, z. B. durch lokale Apps (Bezahl-App, ...)
- Kooperative Entwicklung
  - „von Bibliotheken für Bibliotheken“
  - Kooperation mit kommerziellen Partnern
- Erste Migrationspläne von OLE-Partnerbibliotheken (Texas A+M, Chicago, Cornell) bereits für Sommer 2019
- Anforderungen von Bibliotheken an Datenhoheit, Datensicherheit und Selbstbestimmtheit werden erfüllt



# FOLIO – Geschäftsmodelle und Einsatzszenarien

- Freie Wahl des Geschäftsmodells
  - Lokale Instanz
  - Kommerzielles SaaS-Angebot (Cloud), z. B. von EBSCO, Index Data, ...
  - SaaS / Hostingangebote von Service Providern aus dem öffentlich-rechtlichen Bereich, s.u.
- Szenario: Verbundzentralen-Hosting
  - Ressourcenbündelung durch Kooperation: Services, Entwicklungen und Pflege
  - Individualisierung und Lokalisierung
  - Projekt-Expertise durch Mitarbeit von Anfang an

FOLIO  
—  
Community





# FOLIO – Special Interest Groups

## SIGs

- Metadata Management
- Resource Access
- Resource Management
- User Management
- Internationalization
- Consortia
- Reporting
- Privacy
- Accessibility
- System Operations and Management

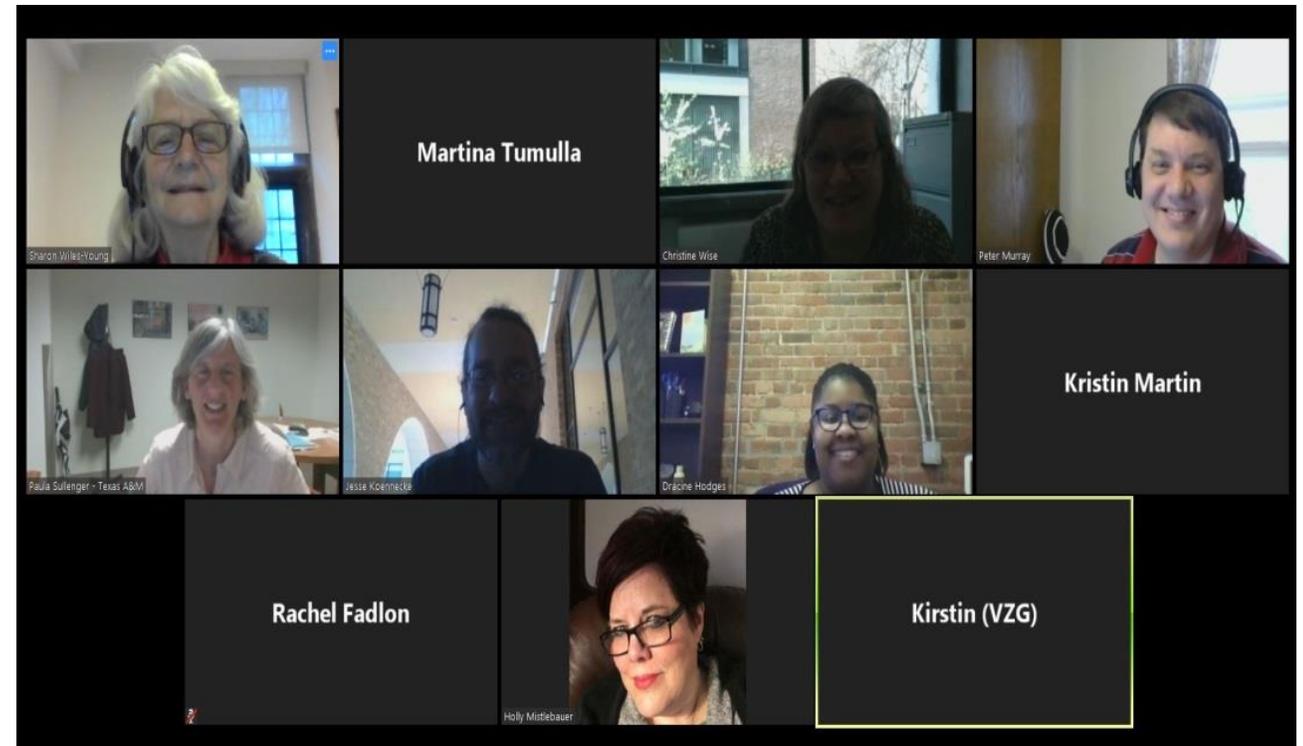
## Über die SIG

- Jeweils zu einem Modul / Fachgebiet / Fragestellung
- Besetzt mit Fachexperten aus der bibliothekarischen Praxis
- Betreut durch einen Convener und/oder Product Owner
- Berichtspflichtig gegenüber dem FOLIO Product Council
- Wöchentliche Webkonferenzen
- Definieren Anforderungen und Anwendungsfälle gemeinsam in der SIG



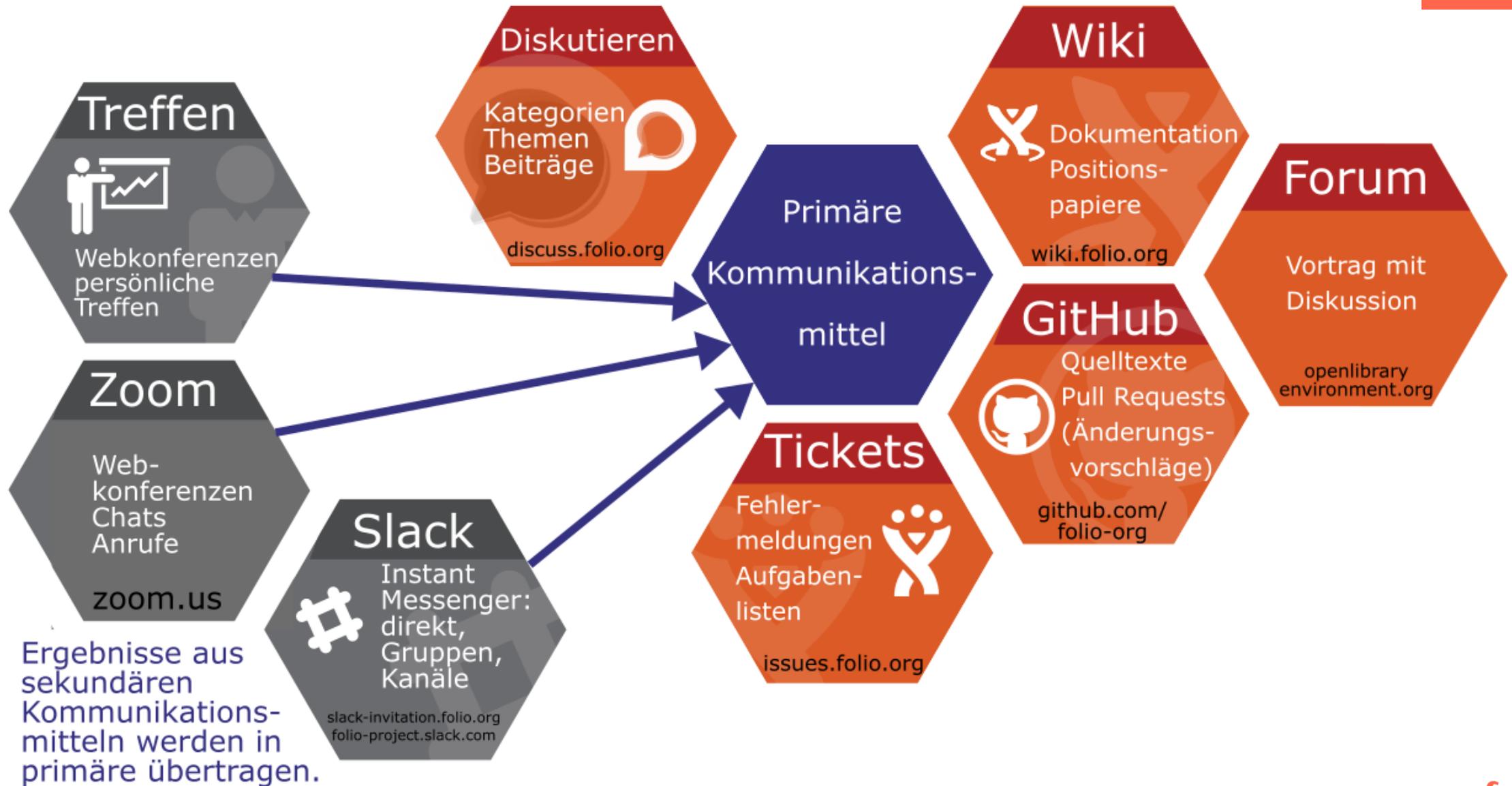
# FOLIO – Product Council

- Koordiniert und begleitet die Arbeit der SIGs
- Informationsknotenpunkt
- Verantwortlich für die Roadmap, Priorisierung und Ressourcenverteilung
- Aktuelle Besetzung aus OLE-Partner-Bibliotheken, EBSCO und Index Data
- Wöchentliche Webkonferenzen





# FOLIO – Kommunikation



FOLIO

—

# Community-Aufbau in Deutschland





# FOLIO-Team in hbz und VZG

- Aktive Mitarbeit im FOLIO-Projekt
  - Funktionale Experten und Entwickler
  - FOLIO Product Council, Stakeholders, SIGs, Subgroups
- Aktive Mitarbeit in der OLE-Community
  - Board, Steering Committee
- Innen- und Außenvertretung des Projektes
  - Deutsche Website (<https://www.folio-bib.org/>)
- Kontinuierliche Evaluation von Software und Anforderungen während des Entwicklungsprozesses



# FOLIO – ERM

- Eigenes Entwicklungsprojekt von GBV/hbz seit März 2018
  - Electronic Resource Management-Funktionalitäten
- Finanzierung durch GBV (Verbundleitungsbeschluss vom 30.11.2017)
  - Entwicklerfirma Knowledge Integration (K-Int, <https://www.k-int.com/>)
  - Product Owner, UX/UI Designer
- Zusammensetzung der ERM Subgroup:
  - Entwickler: OLE/Mellon, hbz, VZG, UB Leipzig
  - Funktionale Expertise: Resource Management SIG, FOLIO-Team GBV/hbz, UB Leipzig
  - Mitarbeit weiterer deutscher Verbände und Bibliotheken

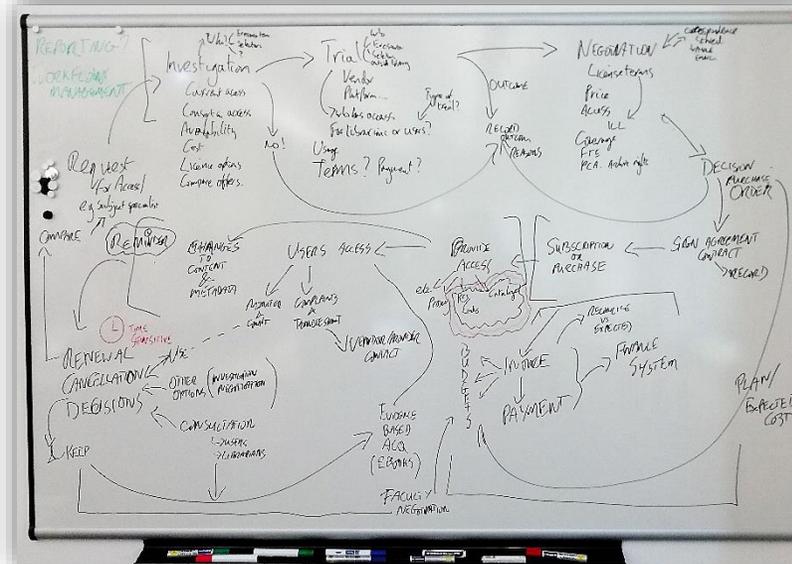


# FOLIO – ERM

- Bereitstellung von ERM-Funktionalität in FOLIO:
  - Paketmanagement
  - Lizenzmanagement
  - Zugangsmanagement



 Einbettung in die Roadmap und den FOLIO Core





# Aufbau einer deutschen FOLIO-Community

- 2014 Beginn des Kooperationsprojektes von hbz und VZG
- Seit 2016 im FOLIO-Projekt und Aufbau einer deutschen FOLIO-Community
- Workshops, Vorträge, Informationstage, ERM-Entwicklungsprojekt
- Aktueller Stand:
  - Deutsche FOLIO-Community wächst rasant → Beteiligung aus **fünf** Verbänden
  - Beitrag: Ressourcen für FOLIO-Projekt (Entwickler und Fachexperten)



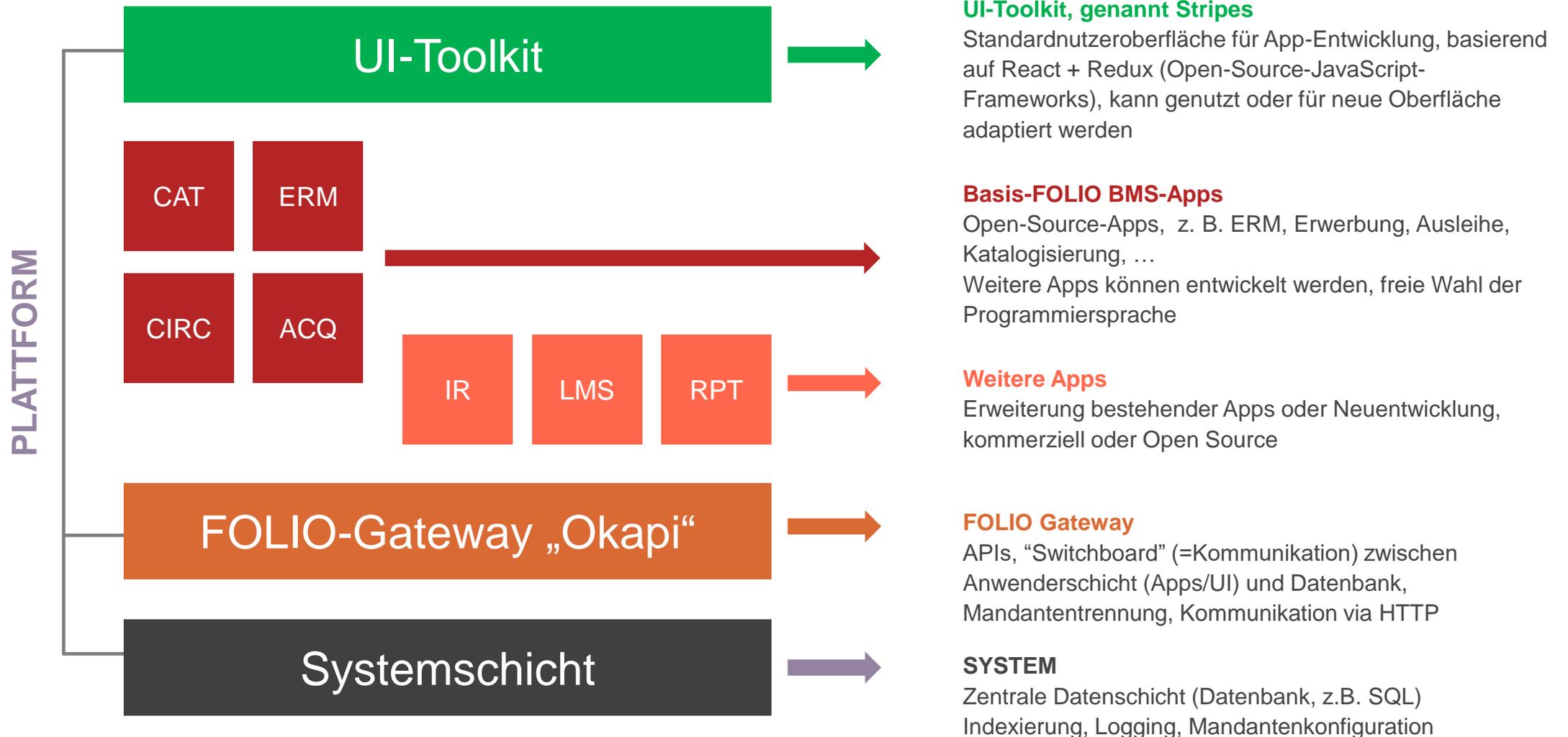
**Seit Mai 2018 neuer deutscher OLE-Partner:**

UNIVERSITÄTS  
BIBLIOTHEK *ubl* LEIPZIG

FOLIO  
—  
Technisches  
Konzept



# Architektur





# Technisches Konzept

- Offene Plattform: Library Service Platform (LSP)
- Plattform stellt Infrastruktur für funktionale Module bereit
- Funktionale Apps  $\Leftrightarrow$  eigenständige Programme
  - Können unabhängig voneinander entwickelt werden
  - Können einzeln ausgewählt und installiert werden
  - Kommunikation über Schnittstellen
- Flexibel erweiterbar, modular („Plug and Play“)
- Mandantenfähig, cloudfähig



# Technisches Konzept

- Design orientiert sich an Microservice-Idee
- „Durchgängig APIs“ bedeutet, dass
  - jeder Entwickler mit jeder Schicht in der Plattform interagieren kann, aber nur über die Schnittstellen (APIs),
  - keine Komponente zu groß ist, um sie zu ersetzen, und
  - freie Wahl von Programmiersprache und Lizenzmodell
- Basierend auf heutigen Anforderungen mit Ausrichtung auf zukünftige Bedürfnisse



# Technologien

Moderner Softwarestack aus bewährten Komponenten

## Frontend (= im Browser)

- JavaScript (ECMAScript 6)
- React/Redux

## Backend (= auf Server)

- Java 8
- Vert.x (asynchrone Kommunikation)
- RAML
- PostgreSQL
  - JSONB (NoSQL) und
  - relationales SQL



# React und Redux

- Sind Open-Source-JavaScript-Webframeworks für Single-Page-Applications (SPAs) = Einseitenwebanwendungen
- React bietet ein Grundgerüst für die Ausgabe von User-Interface-Komponenten in HTML
- Redux ist ein Datencontainer, vereinfacht Lesen vom und Schreiben zum Backend
- <https://reactjs.org/> und <https://redux.js.org/>



# Stripes

- JavaScript-Programm-bibliothek für Frontendmodule
- Basiert auf React + Redux
- Zugeschnitten auf Okapi
  - Kommunikation via Okapi zu Backendmodulen
  - Granulare Nutzerrechte
  - Locale (Sprache, Datumsformat, ...)
  - Logging via Okapi
- <https://github.com/folio-org/stripes-core/#readme>



# vert.x

- Bibliothek für Java
- Ermöglicht einfache Nebenläufigkeit
  - Umgeht viele Probleme paralleler Programmierung
  - Asynchrone Kommunikation
  - Reaktive Programmierung
  - Entwurfsmuster „Reactor“
- <https://vertx.io/>



# RAML

- RAML = RESTful API Modeling Language
- Schnittstellenbeschreibung der Module
- Daraus wird automatisch erzeugt:
  - Schnittstellendokumentation, siehe <https://dev.folio.org/doc/api/>
  - Java-Code (Interfaces)
  - Validierung, die Okapi beim Schnittstellenaufruf durchführt:
    - Erforderliche Benutzerrechte vorhanden? Datenformat korrekt?
- <https://github.com/folio-org/raml-module-builder>

FOLIO  
—  
Software-  
entwicklung





# Datenbankauswahl

- 2016 Tests mit PostgreSQL und MongoDB
- Wahl fiel auf PostgreSQL. Grund: unterstützt gleichzeitig (!)
  - sowohl relationales SQL-Datenbankmodell
  - als auch dokumentenbasiertes NoSQL-Datenbankmodell
- NoSQL = Not-only-SQL, in diesem Fall dokumentenbasiert
- PostgreSQL kann JSON-Dokumente als JSONB verarbeiten, also in einem effizienten binären Format: Das JSON-Dokument wird in seine Bestandteile zerlegt und dadurch indexierbar gemacht
- FOLIO speichert die meisten Daten als JSONB
- Hochverfügbarkeit und Replikation durch PostgreSQL-Cluster

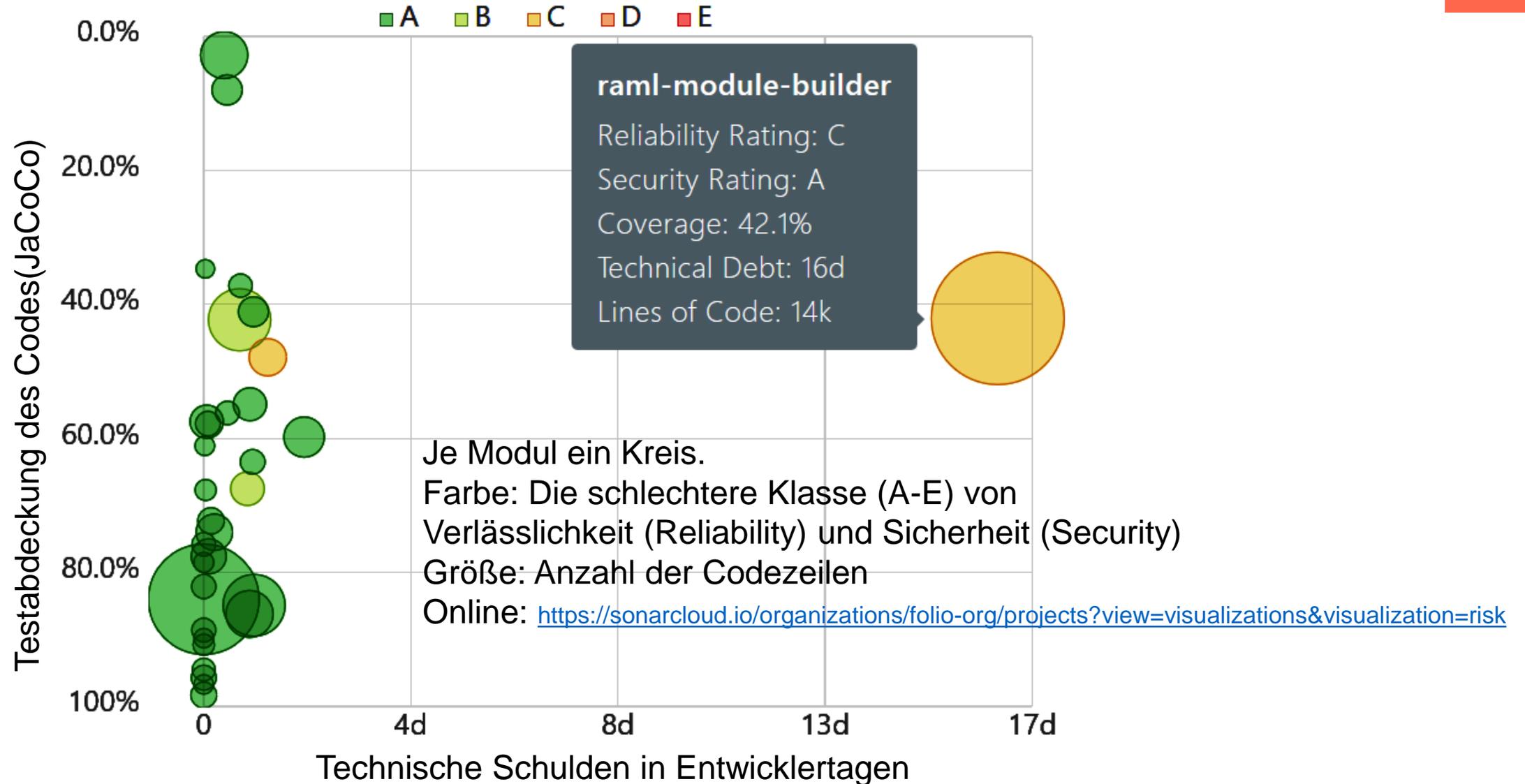


# Mandantentrennung

- Mandant = völlig unabhängige Institution  
(Institutsbibliothek kein Mandant, sondern hat granulare hierarchische Zugriffsrechte)
- Je Mandant
  - eine eigene logische Datenbank (PostgreSQL „Schema“)
  - mit eigenem Datenbanknutzer (PostgreSQL „Role“)
- Datenbankverbindung mit Datenbanknutzer, PostgreSQL garantiert Mandantentrennung



# Codeanalyse der Module





# Qualitätssicherung

- Code-Review durch andere Softwareentwickler
- Automatische Tests (Unit Test, Performance, Codeanalyse, ...)
- Akzeptanztests jeder Anforderung sowie Systemtests
  - durch die SIGs, also durch Bibliothekare bzw. Experten für Barrierefreiheit, Datenschutz, usw.
- Mehrere Testinstallationen
  - Testinstallationen von hbz und VZG vor allem für deutschland- und verbundspezifische Anforderungen
- Erfolgreiche technische Evaluation bezüglich Nachhaltigkeit

FOLIO  
—  
Fazit





# Fazit

- FOLIO ist mehr als ein Bibliotheksmanagementsystem!
- Im Fokus sind die offene Community und die offene Plattform
  - Modulweise Migrationsmöglichkeit limitiert Risiken
  - Offene Schnittstellen intern und extern
  - Aktuelle Software
  - Vereinbarkeit von öffentlichen und kommerziellen Interessen



Nachhaltige Weiterentwicklung bibliothekarischer Funktionalitäten und Services in selbstgewählter Verantwortung unter einer Oberfläche.



# Fragen?

Maike Osters, hbz, Leitung hbz-FOLIO-Projekt  
[osters@hbz-nrw.de](mailto:osters@hbz-nrw.de)

Julian Ladisch, VZG, FOLIO-Softwareentwicklung  
[julian.ladisch@gbv.de](mailto:julian.ladisch@gbv.de)



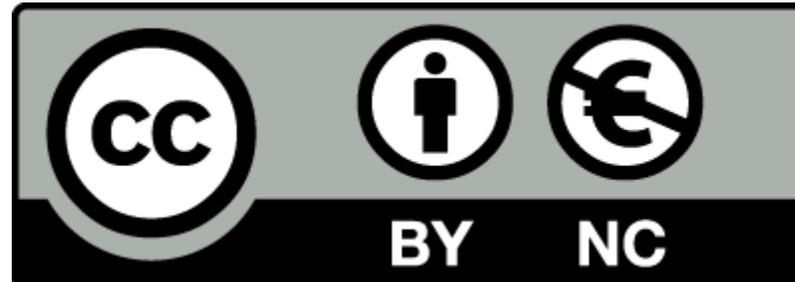
# FOLIO – Links

- Deutsche FOLIO-Website  
<https://www.folio-bib.org/>
- Demoinstallation  
<https://folio-demo.gbv.de/>  
(diku\_admin / admin)
- User-Interface-Prototypen  
<http://ux.folio.org/>
- Code auf GitHub  
<https://github.com/folio-org>



# FOLIO – Links

- Einstiegsinformationen – FOLIO-Wiki  
<https://wiki.folio.org/>
- OLE-Community  
<https://www.openlibraryenvironment.org/>
- Open Library Foundation  
<http://www.openlibraryfoundation.org/>
- FOLIO Allgemein  
<https://www.folio.org/>



Der Text dieser Präsentation wird unter der Lizenz Creative Commons Namensnennung-Nicht kommerziell 4.0 International (CC BY-NC 4.0) veröffentlicht: <https://creativecommons.org/licenses/by-nc/4.0/>

Davon ausgenommen sind die verwendeten, nicht von den Autoren erstellten Grafiken, Screenshots und Bilder, deren jeweilige Rechte und Lizenzbedingungen fortgelten.