INSTITUT FÜR INFORMATIK

Publikationsprozesse für Forschungsdaten mit PubFlow: Von der Erhebung und Verarbeitung zur Archivierung und Publikation

Wilhelm Hasselbring, Marc Adolf, Peer Brauer, Claas Faber, Stefan Farrenkopf, Hela Mehrtens, Arnd Plumhoff, Guido Scherp, Barbara Schmidt, Ralf Schultze, Thorsten Wetzenstein

Bericht Nr. TR_1704

Dezember 2017

ISSN 2192-6247

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

ZU KIEL

Institut für Informatik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Publikationsprozesse für Forschungsdaten mit PubFlow: Von der Erhebung und Verarbeitung zur Archivierung und Publikation

Wilhelm Hasselbring, Marc Adolf, Peer Brauer, Claas Faber, Stefan Farrenkopf, Hela Mehrtens, Arnd Plumhoff, Guido Scherp, Barbara Schmidt, Ralf Schultze, Thorsten Wetzenstein

Bericht Nr. TR_1704
Dezember 2017
ISSN 2192-6247

e-mail: hasselbring@email.uni-kiel.de

Project Report

Zusammenfassung

Die Ergebnisse des DFG-geförderten Projektes PubFlow werden präsentiert. PubFlow zielt darauf ab, Publikationsprozesse für Forschungsdaten von der Erhebung und der Verarbeitung bis hin zur Archivierung und Publikation zu unterstützen. Die exemplarische Implementierung von PubFlow orientiert sich an etablierten Arbeitsabläufen des Forschungsdatenmanagements in den Meereswissenschaften.

1 Einleitung

Die Bedeutung von Datenbeständen nimmt für die wissenschaftliche Arbeit stetig zu. Insbesondere ein vereinfachter Zugang zur Erhebung, Verarbeitung und Veröffentlichung dieser Daten ist hierbei ein wichtiger Aspekt. Um Daten, die im wissenschaftlichen Arbeitsprozess entstehen oder verwendet werden, nachnutzbar zu gestalten ist es erforderlich diese zu publizieren. So können Ergebnisse leichter nachvollzogen und auf den vorhandenen Daten aufgebaut werden. Der Nutzen von Datenpublikationen zeigt sich in vielerlei Aspekten. Beispielsweise können die Daten aggregiert werden, um neue Erkenntnisse zu erlangen. Aufgrund der geteilten Verantwortung für die Datenakkumulation, ist es einfacher große Mengen an aktuellen Daten als Grundlage für eigene Forschungsaktivitäten zu gewinnen [Kel+15]. Weiterhin hat die Datenpublikation direkte Auswirkungen auf Autorinnen und Autoren. Studien zeigen, dass Arbeiten, die zusammen mit ihren Daten veröffentlicht werden, öfter zitiert werden als vergleichbare Paper ohne Daten [PDF07].

Für die Überprüfung der publizierten Erkenntnisse und deren Reproduktion stellen Datenpublikationen wie auch die Veröffentlichung von Software-Code eine unabdingbare Voraussetzung dar. [DTA86] haben bereits 1986 für die empirischen Wirtschaftswissenschaften ernstzunehmende Hinweise auf die Nichtreproduzierbarkeit von Forschungsergebnissen veröffentlicht, die vielfach von Studien neueren Datums, auch für andere Fachrichtungen, bestätigt wurden [AW08; Sci11].

Der Prozess zur Forschungsdatenpublikation besteht aus mehreren Teilschritten. Am Anfang steht die Erhebung der Daten. Diese werden in so genannten *Scientific Workflows* verarbeitet und analysiert. Am Ende des Vorgangs werden die Daten bei Forschungsdatendiensten oder *World Data Center*, wie *Pangaea*[1], publiziert. Diese ermöglichen einen einfachen Zugang, vergeben eindeutige, zitierfähige Identifikatoren und archivieren Forschungsdaten.

Eine wesentliche Voraussetzung für PubFlow stellt die Etablierung des Kieler Datenmanagementteams[2] dar, welches als gemeinsame Einrichtung des GEOMAR, des Exzellenzclusters „Ozean der Zukunft“, sowie der Sonderforschungsbereiche 574 (Fluide und Volatile in Subduktionszonen: Klima – Rückkopplungen und Auslösemechanismen von Naturkatastrophen) und 754 (Klima – Biogeochemische Wechselwirkungen im tropischen Ozean) gebildet wurde.

Der primäre Beitrag des PubFlow-Projektes besteht in der workflowbasierten Verknüpfung der temporären Datenhaltung (z.B. in lokalen Repositorien) und der Datenarchivierung und -publikation aus dem Blickwinkel wissenschaftlicher Einrichtungen. Für die Datenverarbeitungsprozesse kann auf umfangreiche Vorarbeiten zurück gegriffen werden. Im D-Grid-I–Projekt WISENT[3] wurden Scientific Workflows für auf dem Weather Research and Forecasting Model (WRF)[4] [Has+06; Has09] basierende Wettervorhersagen im D-Grid [Plo+09] entwickelt. Dabei gibt es eine Pre-, Haupt- und Postprozessierungsphase, sowie eine anschließende Visualisierung. Im D-Grid-II–Projekt BIS-Grid[5] [Has10; Gud+08b] wurde mit der BIS-Grid Workflow En-

[1] https://www.pangaea.de
[2] https://www.geomar.de/en/service/data-management/
[3] http://wisent.d-grid.de
[4] http://www.wrf-model.org/
[5] http://bisgrid.d-grid.de

gine[6] eine auf WS-BPEL[7] basierende, Grid-kompatible Workflow-Engine entwickelt, die sowohl Business Workflows als auch Scientific Workflows sicher und zuverlässig ausführen kann [SH11; SH10; Gud+10; Sch+10; Gud+08c; Gud+08a].

In Abschnitt 2 wird zunächst über die Entwicklung des ersten PubFlow-Prototypen berichtet, bevor in Abschnitt 3 in PubFlow entwickelte Scientific Workflows exemplarisch vorgestellt werden. Die Architekturmodernisierung von PubFlow hin zu einer Microservice-Architektur, die auf Basis der Erfahrungen mit dem ersten Prototypen erfolgte, wird in Abschnitt 4 vorgestellt. Konzepte zur persistenten Identifizierung von Autoren in PubFlow werden in Abschnitt 5 präsentiert. In Abschnitt 6 wird über die Erfahrungen mit den eingesetzten Systemen berichtet, um dann in Abschnitt 7 diesen Bericht zusammenzufassen.

2 Entwicklung des ersten PubFlow-Prototypen

In der ersten Projektphase von PubFlow wurde zunächst ein Konzept geschaffen. Außerdem wurde eine Pilotanwendung für die Meereswissenschaften am Standort Kiel umgesetzt. Dies wurde mit dem Ziel umgesetzt, eine flexible und verlässliche Plattform zu bieten, mit der die Forschungsdaten primär für die eigene Organisation verwaltet werden. Die Langzeitarchivierung ist über eine Anbindung des World Data Center for Marine Environmental Sciences (WDC-MARE, Pangaea) realisiert.

Begonnen wurde mit einer Anforderungsanalyse um zu klären, welche Funktionalitäten das spätere Produkt bieten muss, um die Akzeptanz des Produktes zu maximieren. In diese Analyse sind die Vorarbeiten des Kieler Datenmanagementteams eingeflossen, welches ähnliche Arbeiten bereits für deren sogenannte OCN-Plattform durchgeführt hatte. Zeitgleich wurden verschiedene Workflow-Plattformen evaluiert um zu ermitteln, welche der Plattformen für das Projekt am geeignetsten ist.

Basierend auf diesem Entwurf wurde mit der Bereitstellung der technischen Infrastruktur für das Projekt PubFlow und der Implementierung des Prototypen für ein erstes Evaluationsbeispiel begonnen. Die aktuelle Version des Prototypen implementiert u. a. einen Workflow zum Transfer sogenannter Bottle-Daten aus einem Instituts-Repository des GEOMAR zu Pangaea. Dabei handelt es sich um Messdaten einer CTD-Rosette.[8] Ein Beispiel für einen solchen Workflow ist in Abbildung 1 dargestellt (siehe auch Abschnitt 3).

Die in der Datenbank enthaltenen Daten werden dabei den einzelnen Mess-Events zugeordnet. Die Beschreibung der Methoden und die Parameter-Bezeichnung werden auf die Pangaea-Parameter abgebildet und dann in ein Austauschformat übertragen, in Form von *4D-Files*. Neben diesem Prototypen wurde wurde ein technischer Prototyp für ein konfigurierbares Framework zum Monitoring von Business Workflows entwickelt [Bra14; BFH14]. Dieses Monitoring-Framework erlaubt es, Zustandsänderungen eines Workflows zu erfassen, der in einer Workflow-Engine ausgeführt wird. Dieses Framework soll eingesetzt werden, um während der Ausführung eines Workflows anfallende Provenienz-Informationen automatisch zu erfassen und diese zu aggregieren. Die umgesetzten Workflows sind Scientific Workflows. In PubFlow umfassen diese die Datenerhebungs- und -publikationsprozesse.

Zu Beginn des Projektes hat es sich in Gesprächen mit den assoziierten Partnern als sehr wichtig herausgestellt, die Komplexität der Workflows vor den Nutzern des System weitgehend zu verbergen. Es wurden daher verschiedenen Nutzerrollen (z.B. Datenmanager, Wissenschaftler der Fachdomäne, ...) verschiedene Sichten auf das System zugeordnet. Wissenschaftlern der Fachdomäne wurden die Funktionen des PubFlow-Systems in Form eines Plugins angeboten, das in das Ticketsystem Jira am GEOMAR integriert wurde. Datenmanager hingegen konnten dort neue Publikationsworkflows anlegen und bestehende Workflows bearbeiten. Weiterhin wurden verschiedene Workflow-Plattformen evaluiert, auf denen das PubFlow-System realisiert

[6] http://bis-grid.sourceforge.net/
[7] https://ode.apache.org/ws-bpel-20.html
[8] http://de.wikipedia.org/wiki/CTD-Rosette

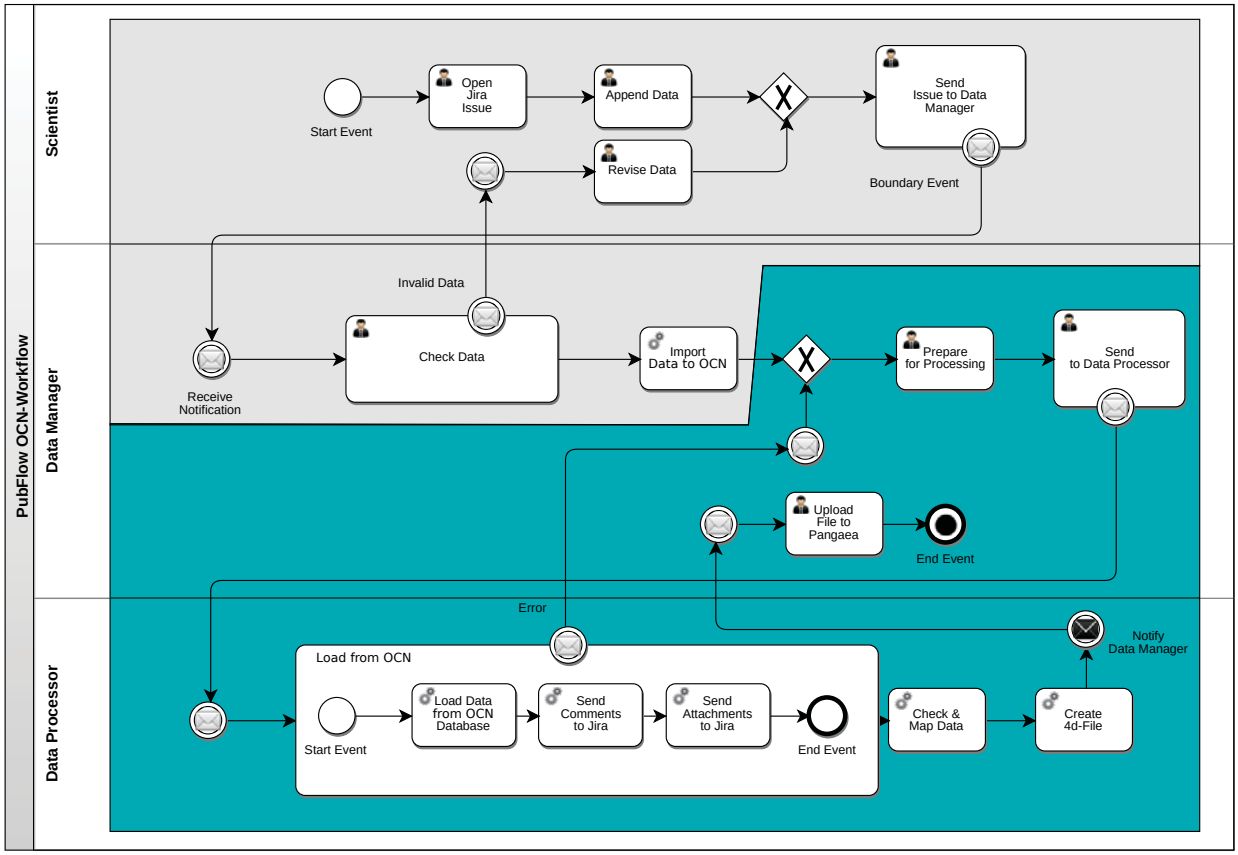

Abbildung 1: Datenpublikation von Daten aus der OCN-Datenbank auf PANGAEA.

werden sollte. Hier zeigte sich, dass die Apache ODE BPEL Engine am geeignetsten ist. Die Gründe hierfür waren zum einen, dass sich ODE nahtlos in den Apache ServiceMix integrieren lässt und es sich zum anderen um einen erprobten Industriestandart handelt. In Abbildung 2 ist die Architektur [Has06] des ersten Prototypen von PubFlow während der ersten Projektphase dargestellt. Hierbei handelt es sich um eine service-orientierte Integrationsarchitekur [Con+05; Has00].

Die erste prototypische Realisierung des PubFlow-Frameworks wurde in den ersten zwei Projektjahren erfolgreich realisiert und bereits durch das Datenmanagementteam am GEOMAR genutzt und evaluiert. In der darauf folgenden Version wurde die Nutzeroberfläche durch die Einbindung des Ticketsystems Jira[9] ersetzt um die Akzeptanz des Tools zu fördern. Die Nutzer und Nutzerinnen interagieren mit PubFlow nur über die Tickets. Diesen wird immer ein Workflow zugeordnet und Zustände zwischen denen mittels vorgegebener Transitionen gewechselt werden kann. An bestimmten Punkten bzw. Zuständen im Workflow werden automatisierte Prozesse angestoßen, zum Beispiel um Datensätze aus lokalen Datenbanken zu laden.

3 Scientific Workflows in PubFlow

In der ersten Projektphase von PubFlow wurde der vom GEOMAR zur Verfügung gestellte Anwendungsfall diskutiert und analysiert. Dieser bestand im Wesentlichen aus der Veröffentlichung von Daten, die im Rahmen von zwei Projekten gesammelt wurden und beim GEOMAR lokal gespeichert waren, im Weltdatenzentrum Pangaea. Dabei werden die Daten aus der lokalen Datenbank entnommen (zunächst aus der *OCEAN*-Datenbank, kurz OCN, später aus der *Cap*

[9] https://www.atlassian.com/software/jira

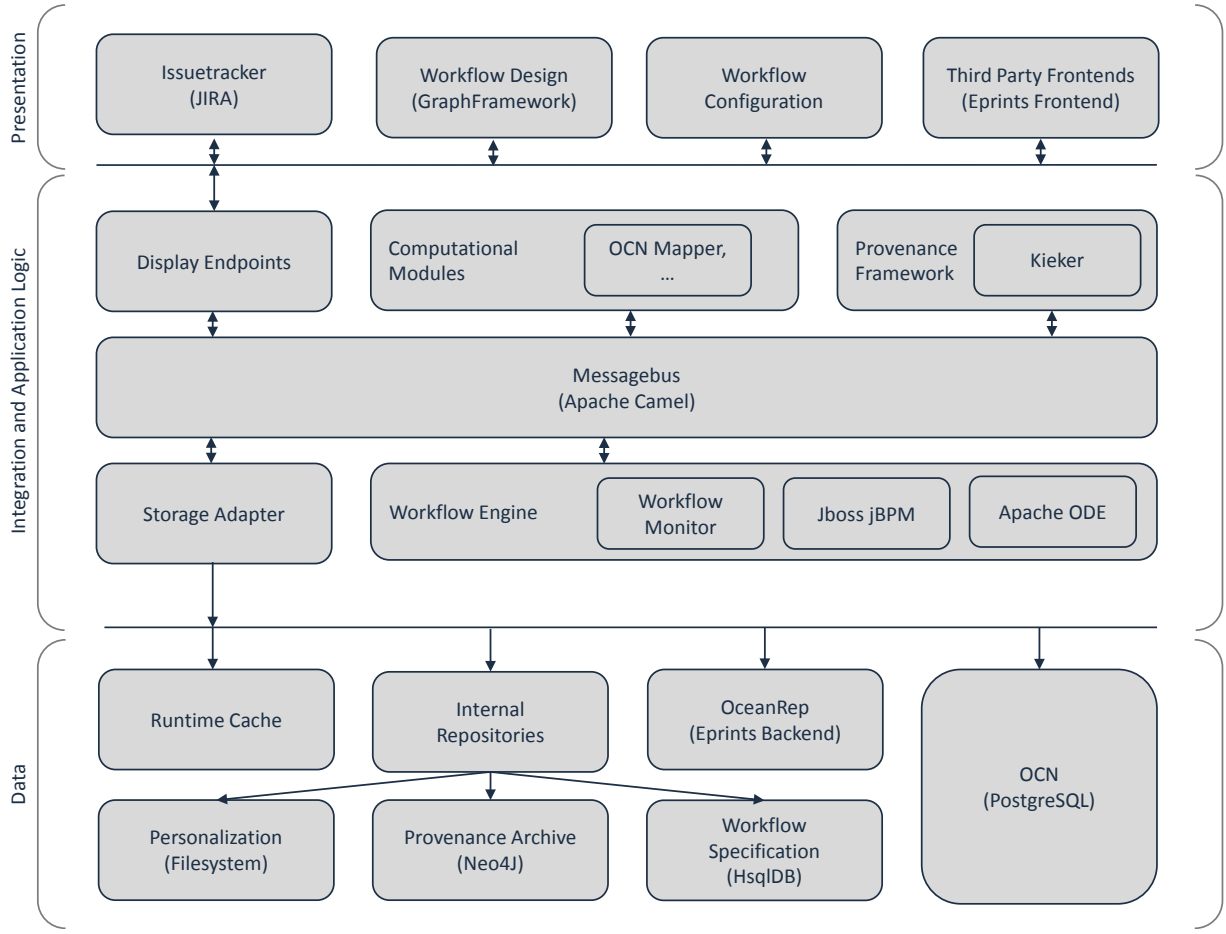

Abbildung 2: Architektur von PubFlow während der ersten Projektphase

Verde Ocean Observatory-Datenbank, kurz CVOO) und für die Publikation vorbereitet. Am Ende wird eine Datei erzeugt, die dem Datenformat für den 4D-Client von Pangaea entspricht, und diese wird dann interaktiv über den Client hochgeladen. In Abbildung 1 ist das Ergebnis dieser Modellierung abgebildet. Dieser Vorgang wurde zunächst, wie abgebildet, als ein großer Workflow betrachtet und implementiert. Später wurde er in zwei zeitlich versetzte Workflows aufgeteilt. Der erste Teil beinhaltet die Übermittlung der Daten vom Wissenschaftler an die institutsinterne Datenbank und die Bereitstellung von Metadaten. Er wird von den verantwortlichen Wissenschaftlern und Wissenschaftlerinnen angestoßen. Der zweite Teil beinhaltet die eigentliche Datenpublikation, die von Datenmanagerinnen und Datenmanagern angestoßen wird.

Die Modellierung in Abbildung 1 stellt alle Schritte des Workflows dar, wobei nur der Teil des Modells, der von der Rolle *Data Processor* übernommen wird, von der BPMN-Engine automatisiert ausgeführt wird. Alle anderen Schritte sind in den Workflows in der Jira-Oberfläche implementiert oder sind manuelle Aufgaben, die sich in diesen Zuständen ergeben.

Der Vorgang beginnt mit einer Nutzerin oder einem Nutzer aus der Gruppe *Scientist*. Sie eröffnen ein Ticket im Ticketmanagementsystem Jira. In diesem Ticket ergänzen sie ihre Daten und entsprechende Metadaten, wie z.B. die zugehörige Fahrt oder das Projekt. Sobald dies erledigt ist, übermitteln sie das Ticket den Mitgliedern der Gruppe *Data Manager*. Die *Data Manager* unterziehen die übermittelten (Meta-)Daten einer ersten Prüfung. Bei Unklarheiten wird mit den Mitteln des Ticketsystems nachgefragt und der Status entsprechend zurückgesetzt. Wenn die Qualitätskriterien eingehalten wurden, werden die Daten in die lokale Datenbank übernommen, hier die OCN-Datenbank. Abbildung 3 zeigt wie die Metadaten über die Eingabemaske von Jira gesammelt werden. Diese Daten werden dann intern von Jira mit dem Ticket gespeichert und für

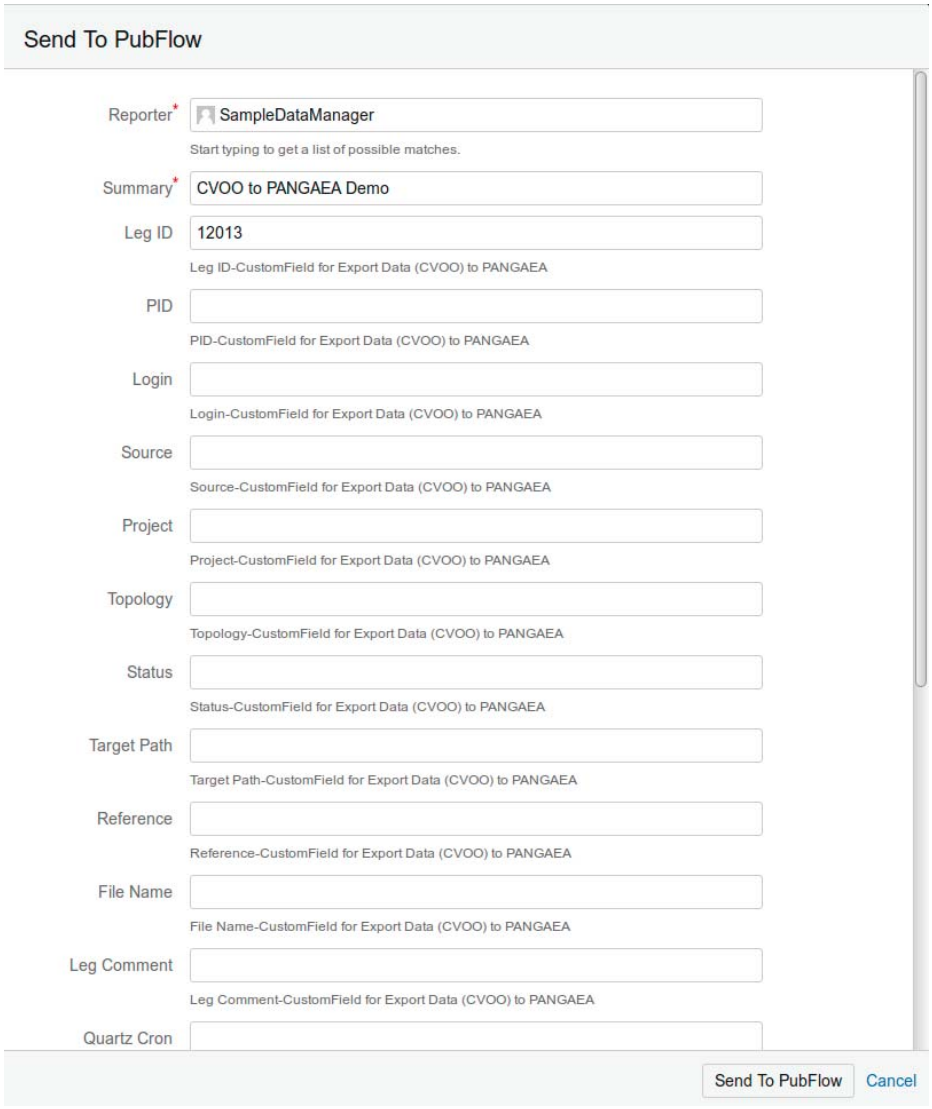

Abbildung 3: Eingabemaske für Metadaten in Jira.

die automatisierte Verarbeitung später ausgelesen. Nicht alle Felder müssen ausgefüllt werden um ein valides Ergebnis zu erhalten.

Nach dieser ersten Phase beginnt der eigentliche Publikationsprozess, der gleichzeitig auch der zweite Teil des aufgeteilten Workflows ist. Dieser zweite Teil wird von der Gruppe *Data Manager* gestartet. Hier werden die Daten zunächst manuell um weitere Metadaten angereichert, die nicht in der Datenbank hinterlegt wurden. Sobald die Daten auf diese Weise für die Verarbeitung vorbereitet wurden, werden sie per Knopfdruck an den *Data Processor* übermittelt. Die Rolle des *Data Processor* repräsentiert die automatisierte Datenverarbeitung in PubFlow. In diesem Workflow sind drei wesentliche Prozesse automatisiert, die hintereinander ausgeführt werden. Zunächst werden die Daten aus der Datenbak geladen. Anschließend werden die Daten überprüft und *quality flags* hinzugefügt. Dabei bleiben die Rohdaten erhalten. Potentielle Messfehler, wie z.B. eine negative Salinität, erhalten eine Markierung. Am Ende werden die Metadaten und die Daten aus dem Schritt davor zu einer Datei im 4D-Format aggregiert. In jedem Schritt wird durch Kommentare und Anhänge an das entsprechende Ticket im Ticketsystem der Fortschritt dokumentiert. Sollten Fehler auftreten wird entsprechendes Feedback generiert und der Status des Tickets entsprechend gesetzt. Die erstellte 4D-Datei wird dann an den *Data Manager* geschickt. Diese Datei wird als letzter Schritt über den 4D-Client von PANGAEA in das WDC geladen. Zukünftig soll dieser letzte Schritt automatisiert über eine Web-Service-Schnittstelle

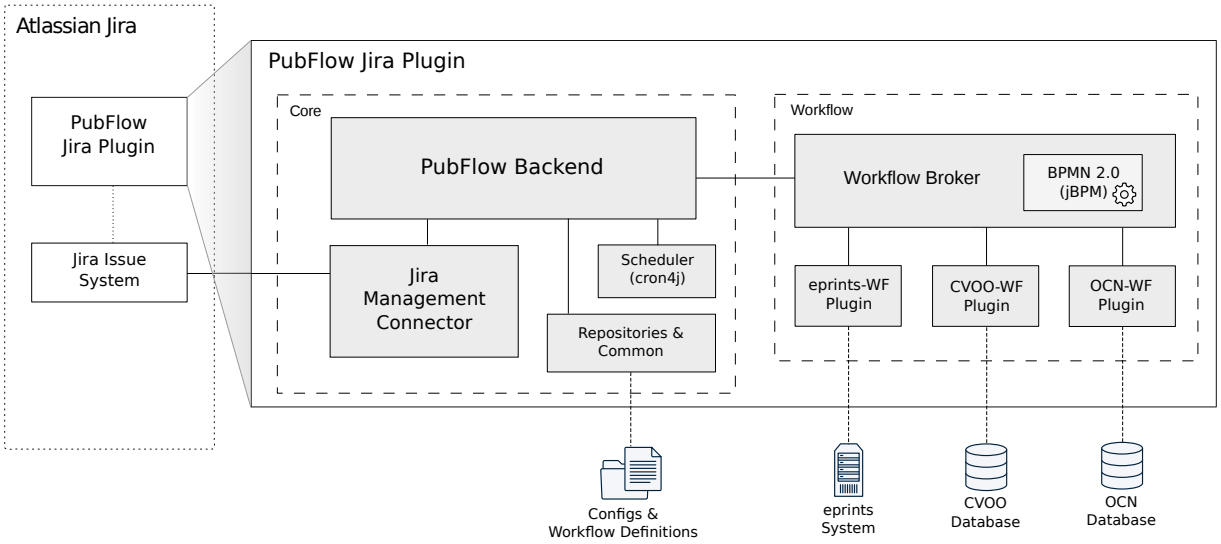

Abbildung 4: PubFlow – Erste Projektphase, Jira-Plugin – monolithische Schichtenarchitektur

erfolgen, die bisher noch nicht zur Verfügung steht, aber in Aussicht gestellt wurde.

Für Datenpublikationen sollten die Daten in allen Schritten um relevante Metadaten angereichert werden, um zum Beispiel deren Herkunft, Verarbeitung, Qualität nachvollziehen zu können. Gleichzeitig müssen sie in die von den Datenzentren vorgegebenen Formate gebracht werden. Auch Maßnahmen, wie Qualitätskontrollen, können dabei auftreten. Häufig werden diese Schritte von den Wissenschaftlern oder Datenkuratoren ausgeführt. Da gleiche Datenarten die gleiche Behandlung erfordern, resultiert dies in repetitiven, häufig manuellen Aufgaben. Das Ziel von PubFlow ist es, Scientific Workflows zu betrachten, die die Teilbereiche der Datenerhebung, der Datenverarbeitung und -analyse, sowie der Datenarchivierung und -publikation unterstützen. Insbesondere soll durch PubFlow die workflowbasierte Datenpublikation ermöglicht werden. Dazu sollten leicht konfigurierbare Workflows genutzt werden. Anwendern sollte es möglich sein, diese anzupassen oder neue zu erzeugen. Das Vorbild war dabei der Industriestandard BPMN.

Zusätzlich zu den bereits erwähnten Metadaten ist auch die Provenienz der Daten wichtig, um ihre Herkunft und Verarbeitung nachvollziehbar darzustellen. In PubFlow wurden hierzu konzeptionelle Arbeiten durchgeführt [BH13a; BCH14; BH13b; Bra12c; BH12] und technische Prototypen entwickelt [Bra14; BFH14; Bra12b; Bra12a].

Abbildung 4 zeigt die installierte Architektur von PubFlow zum Ende der ersten Projektphase. Hier sind alle Bestandteile der Software in einem großen Paket zusammengesetzt, das als Plugin für das Ticketmanagementsystem Jira konzipiert ist. Alle anfallenden Aufgaben werden innerhalb des Plugins verarbeitet. Zusätzlich gibt verschiedene externe Komponenten, wie Datenbanken oder den EPrints-Server des GEOMAR. Diese dienen entweder als Datenquellen oder können über APIs Workflows extern initiieren und bearbeiten.

Innerhalb vom PubFlow-Plugin gibt es zwei Unterteilungen. Zum einen gibt es den ”Core”-Bereich, in dem die Schnittstellen zu Jira und alle damit zusammenhängenden Aufgaben umgesetzt werden. Dies beinhaltet das Anlegen des PubFlow-Projektes und alle seiner zugehörigen Workflows zum System-/Pluginstart. Außerdem werden vom Core die externen APIs zur Manipulation von Tickets innerhalb des PubFlow-Projektes bereitgestellt. Zum anderen ist der ”Workflow”-Teil allein für die Umsetzung und Ausführung der automatisierten Workflows zuständig. Diese Workflows können zum Beispiel aus dem automatischen Lesen von Daten aus einer Datenbank, dem anschließenden Einfügen von *quality flags* und dem Export als Datei bestehen. Sie sind unabhängig von den Workflows der Jira-Tickets implementiert. Der Kontrollfluss geht dabei von den Tickets aus, die über das Plugin die verschiedenen automatisierten Workflows aufrufen. Daraus ergibt sich eine Komposition der Ticket-Workflows in Jira und der automati-

schen BPMN-Workflows. In Jira werden die einzelnen Schritte manuell vom Nutzer ausgelöst und dabei von verschiedenen Rollen mit Informationen angereichert. An vorher definierten Stellen können die automatisierten Abläufe gestartet werden. Die Ergebnisse werden dann direkt im Ticket festgehalten, so dass der Bearbeiter oder die Bearbeiterin über die Oberfläche von Jira den Vorgang kontrollieren und fortführen kann.

4 Architekturmodernisierung von PubFlow

In der zweiten Projektphase wurden die bereits fertig gestellten Komponenten aus der ersten Projektphase von PubFlow weiter genutzt, angepasst und evaluiert. Der erste Arbeitsabschnitt im Projektzeitraum war die Modernisierung der Softwarearchitektur, die im Rahmen des Projektes entwickelt und genutzt wurde. Der Ausgangspunkt unserer Entwicklung war dabei die “monolithische” Architektur aus der ersten Projektphase. Die Einarbeitung des neuen Mitarbeiters, die Planung und der Umbau haben die erste Hälfte des bewilligten Jahres in Anspruch genommen.

Durch die längere Pause zwischen den beiden Projektphasen entstand Modernisierungsbedarf. Insbesondere war es in der inzwischen erschienenen Jira-Version 7 nicht mehr möglich das Plugin der älteren Jira-Version zu verwenden. Eine Überarbeitung des existierenden Plugins war nicht effizient durchführbar [AH16]. Deshalb wurde die Software-Architektur von einer Komponenten-Architektur [Has02] zu einer Microservice-Architektur [HS17; Has16; New15] umstrukturiert. Abbildung 5 zeigt die überarbeitete Microservice-basierte Architektur von PubFlow. Die monolithische Struktur, in der alle Aufgaben innerhalb des Jira-Plugins gekapselt waren, wurde aufgebrochen und auf verschiedene unabhängig installierbare Services verteilt.

In dieser Architektur übernimmt das Jira-Plugin nur die Aufgaben, die zur Verwendung von Jira notwendig sind. Alle weiteren Aufgaben werden über *REST-Schnittstellen* an entsprechende Services verteilt. Die zweite relevante Komponente ist der *Workflow Provider*. Dieses eigenständige Programm bietet verschiedene Schnittstellen an, über die unterschiedliche Services genutzt werden können. Das System ist dabei nicht nur auf einen solchen Provider beschränkt. Es ist möglich, weitere Services zu nutzen. Da jegliche Kommunikation standardisiert über REST erfolgt, können auch externe Services ohne größere Anpassungen verwendet werden. Updates, Statusänderungen, Kommentare, etc. werden über die API des Jira-Plugins in die Tickets übertragen.

Die von uns gestalteten Workflows im *Workflow Provider* werden in der *Business Process Model and Notation* (BPMN) entworfen und abgespeichert. Da die reine Beschreibung der Workflows als BPMN nicht ausreicht, um sie ausführbar zu machen, wurden sie mit Java-Code annotiert. In PubFlow nutzen wir dafür *jBPM*[10]. Diese Workflows beinhalten keine manuellen Aufgaben, da die manuellen Aufgaben über das Ticketsystem von Jira erledigt werden.

Innerhalb der Workflows können wiederum andere Services aufgerufen werden. In unseren Beispielanwendungen geschieht dies um Daten aus Datenbanken (CVOO oder OCN) abzufragen oder Daten zu verarbeiten. Der *EPrints-Workflow Service* andererseits wird nicht von PubFlow aufgerufen, sondern stößt seinerseits das Erzeugen von neuen Tickets im PubFlow-Jira-Plugin an. Zusätzlich werden bestehende Programme, wie Datenbanken oder der EPrints-Server genutzt. Außerdem besteht auch die Möglichkeit einen weiteren Service zu nutzen, um die Kommunikation und Lastverteilung zwischen einzelnen *Service Providern* zu ermöglichen. In der Abbildung ist diese Komponente mit gestrichelten Außenlinien dargestellt und als *Workflow Distributor* gekennzeichnet.

Abbildung 6 zeigt wie die einzelnen Komponenten von PubFlow im echten Betrieb physikalisch am GEOMAR verteilt sind. Die Grafik repräsentiert den Zustand des Systems zum Projektende. Die Komponenten, die in dunkelgrau eingezeichnet sind, stellen die für PubFlow erstellten Komponenten dar. Die in Abbildung 5 diskutierten Teile der Architektur finden sich auch in diesem Diagramm wieder. Jedes in diesem Rahmen erstellte Programm läuft auf dem gleichen Server

[10] http://www.jbpm.org/

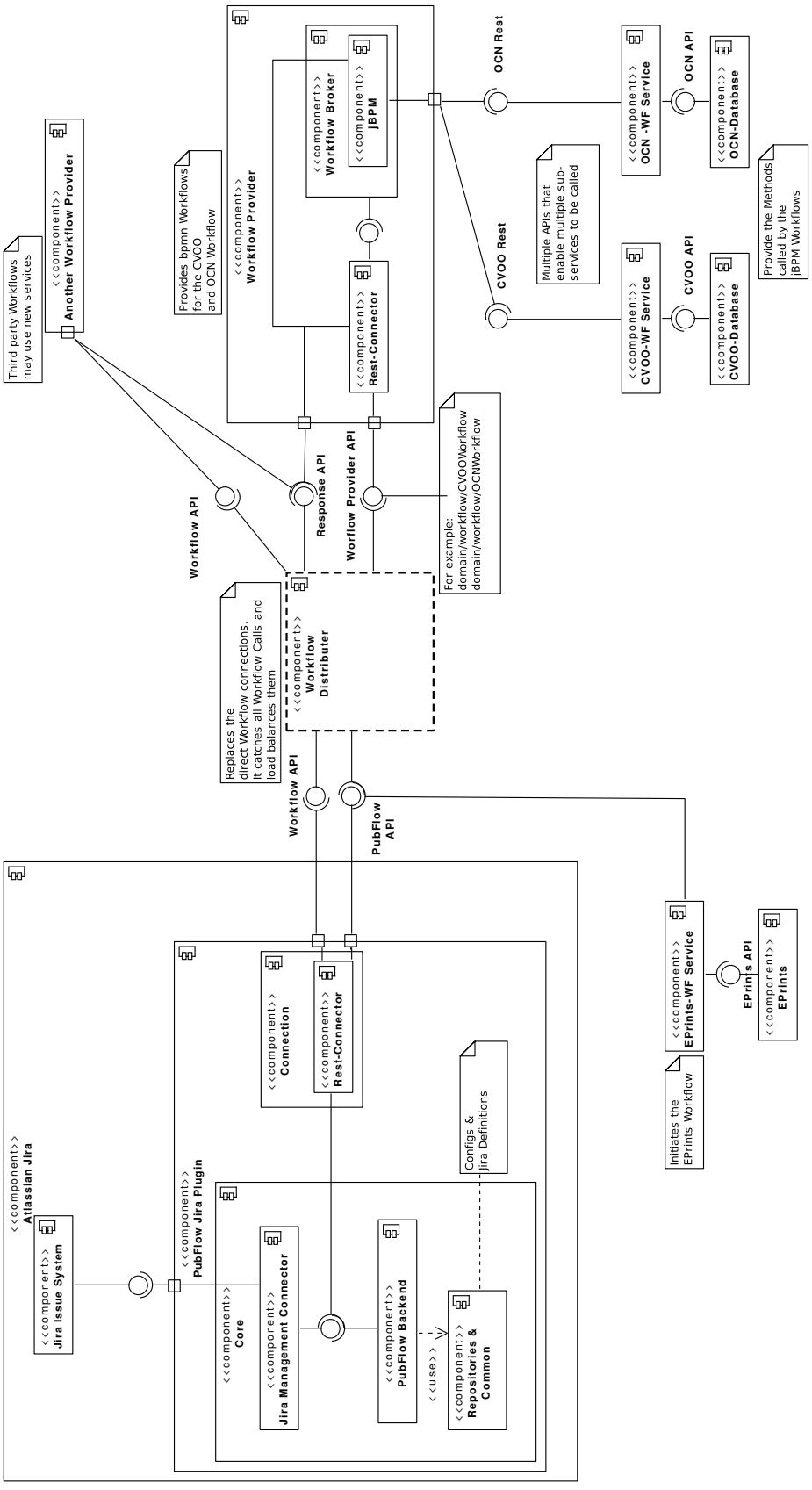

Abbildung 5: PubFlow – Zweite Projektphase – Microservice Architektur

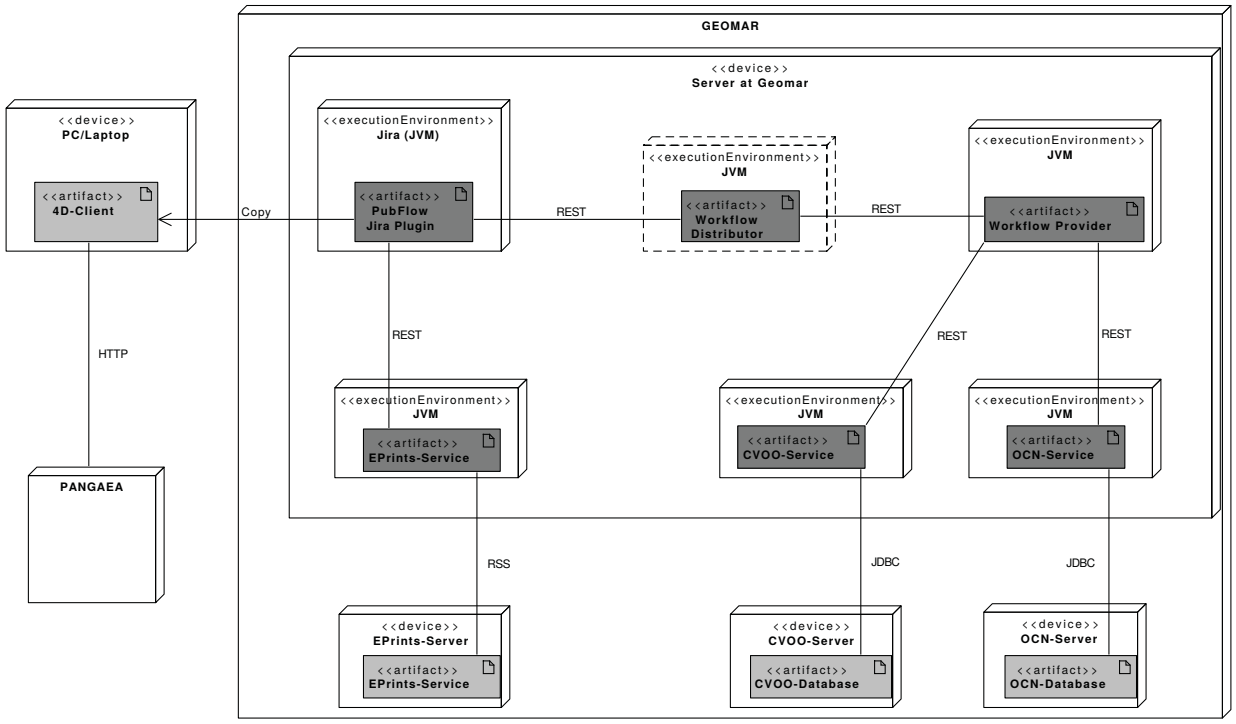

Abbildung 6: PubFlow - Deployment am GEOMAR und Verbindung zu PANGAEA

innerhalb des GEOMAR-Computernetzwerks. Da sie als Microservices konzipiert wurden, könnten sie aber auch auf verschiedenen Servern verteilt installiert werden. Alle Services laufen in einer separaten Laufzeitumgebung. Alle Services beruhen auf Java. Deshalb stellt die Java Virtual Machine (JVM) die Laufzeitumgebung dar. Das PubFlow-Jira-Plugin wird wiederum innerhalb von Jira ausgeführt. Da auch Jira ein Java-Programm ist, wird hier ebenfalls die JVM genutzt.

Der EPrints-Service und die beiden Datenbankserver für die OCN- und CVOO-Datenbank werden nicht auf dem selben Server ausgeführt. Allerdings befinden sie sich auch innerhalb des Netzwerkes vom GEOMAR.

5 Persistente Identifizierung von Autoren in PubFlow

PubFlow zielt darauf ab den Prozess der Veröffentlichung von Forschungsdaten auf geeigneten Repositorien zu unterstützen. Da die Forschungsdaten Grundlage für Publikationen sind, wurde in der zweiten Projektphase das ursprüngliche Vorhaben um den Aufbau bibliothekarischer Dienstleistungen erweitert. Konkret geplant sind die Einführung von persistenten Identifikatoren für Personen sowie die Nachnutzung der beschreibenden Metadaten, die im PubFlow-Prozess entstehen, für die Verknüpfung von Publikationen und zu Grunde liegenden Daten. Darauf aufbauend kann ein Nachweis der Ergebnisse von Forschungstätigkeit in geeigneten lokalen, regionalen oder fachspezifischen Systemen erfolgen. An der CAU und am GEOMAR wurden zum Zeitpunkt der Antragsstellung mit den Repositorien bereits einige lokale Systeme, die Forschungsergebnisse veröffentlichen betrieben. Ein Forschungsinformationssystem (FIS) und die Hochschulbibliographie befinden sich an der CAU im Aufbau. Schnittstellen und Prozesse zum Austausch von Daten zwischen Bibliothek und FIS sind bereits konzipiert und teilweise eingerichtet.

Ein wichtiges Ziel ist die Unterstützung der Autorenidentifikation innerhalb der Prozesse von PubFlow. Mit dieser Funktionalität soll die Zuordnung von Klarnamen bzw. internen Identifikationsmerkmalen zu globalen, eindeutigen Identifikatoren geleistet werden. Durch die Kombination der fachlichen Kompetenz der UB und der angewandten Methoden der Bibliothek des GEOMAR

kommen zwei Identifikationssysteme für das Projekt in Frage.

Die erste Möglichkeit ist die Nutzung der *Gemeinsamen Normdatei(GND)*[11]. Die GND wird von der Deutschen Nationalbibliothek (DNB) verwaltet. Durch die Einbeziehung der Universitätsbibliothek kann direkt auf das System Einfluss genommen werden und ggf. Einträge angelegt oder aktualisiert werden. Allerdings ist die GND vor allem im deutschsprachigen Raum relevant und wird bisher nicht für Forschungsdaten verwendet. Ein GND-Eintrag erfolgt nach den aktuellen Regeln der DNB erst dann, wenn eine Publikation vorliegt.

Forschungsdaten, die bei PANGAEA[12] veröffentlicht wurden, können Autoren zugeordnet werden, um die Herkunft der Daten eindeutig zu identifizieren. Da sowohl das GEOMAR als auch der bisherige PubFlow-Kontext PANGAEA nutzen, bietet sich ORCID[13] als Alternative zur GND an. ORCID besitzt eine wachsende, internationale Nutzergruppe. Gerade bei internationalen Projekten und Kooperationen am GEOMAR erhöht die Nutzung der ORCID die Sichtbarkeit der Autoren. In diesem System gibt es bisher allerdings weder für die Universitätsbibliothek noch für die Bibliothek des GEOMAR eine Möglichkeit zur Kuration der Autorendaten. Jeder Nutzer und jede Nutzerin ist eigenverantwortlich für die Erstellung, Aktualisierung und die Sichtbarkeit der eigenen Daten. Duplikate oder falsche bzw. ungenügende Informationen werden nicht korrigiert. Dies erschwert sowohl die automatisierte, als auch die manuelle Suche.

In unserem Prototypen werden die Autorinnen und Autoren mit ihrer ORCID verknüpft, sofern eine ORCID existiert und zugeordnet werden kann. Die ORCID wird am Ende zu PANGAEA übermittelt und dort in den internen Personendatensatz aufgenommen. Pangaea verknüpft in der Anzeige bei vorliegenden ORCID's die Autorennamen mit dem ORCID-Profil. Abbildung 7 zeigt die erste Erweiterung eines unserer Workflows um die Autorenidentifikation. Der gesamte Vorgang ist in BPMN dargestellt. Für diese Aufgabe wird die neue Rolle des *Librarian* hinzugefügt. Der Grundgedanke dabei ist, dass Angestellte mit bibliothekarischer Fachkompetenz, die geeignetsten Ansprechpartner für die Zuordnung von Namen und Identifikatoren sind.

Am Anfang erstellt eine Datenmanagerin oder ein Datenmanager ein Ticket zu diesem Workflow. Anschließend werden die Namen der Autoren und Autorinnen hinzugefügt, für die eine Zuordnung zu einem Identifikator erforderlich ist. Als nächstes wird das Ticket an die Gruppe *Librarian* weitergeleitet. An diesem Punkt soll die Zuordnung zwischen Namen und Identifikatoren stattfinden. In unserem ersten Ansatz ist dies eine rein manuelle Aufgabe. Sobald dieser Schritt fertig ist, wird das Ticket wieder der Gruppe *Data Manager* zugeordnet. Unsere Erweiterung ist damit abgeschlossen und der restliche Vorgang erfolgt wie bisher. In den nächsten Schritten werden weitere Metadaten zu dem Ticket hinzugefügt und für die automatische Weiterverarbeitung an den *Data Processor* geleitet. Das Ergebnis oder auftretende Fehler werden an die zuständigen Datenmanager und -managerinnen gemeldet. Diese laden die entstandenen Dateien am Ende bei PANGAEA hoch und vermerken unter welcher DOI sie zu finden sind.

Unser Ansatz kann in dieser Form beliebig erweitert werden. Dabei ist es möglich, automatisierte Unterstützung bei der Identifikatorensuche und Zuordnung zu geben. Weiterhin ist es auch denkbar, dass bereits gefundene Zuordnungen abgespeichert und nachhaltig weiterverwendet werden. Dabei stellen sich allerdings Probleme zur Aktualität der gespeicherten Daten ein, zum Beispiel, falls ein Wissenschaftler oder eine Wissenschaftlerin das Institut verlassen hat und eine andere Person mit dem selben Namen bzw. lokalen Identifikator eingestellt wurde.

Die Aufgabe der Autorenidentifikation auf Grundlage lokal gespeicherter Daten erscheint uns als großes Problemfeld, dass noch weiterer Arbeit bedarf. Im Rahmen von PubFlow wurden weitere und auch komplexere Lösungskonzepte konzeptionell entworfen. Mit Blick auf den Datenaustausch und die Nutzung von APIs ist vom Zeitpunkt der Antragstellung und über die Projektphase hinweg eine hohe Dynamik zu beobachten. So ist es heute nicht mehr möglich, dass Einrichtungen für ihre Angehörigen ORCIDs anlegen, sie können aber bestehende Datensätze über eine API pflegen. Es besteht inzwischen die Möglichkeit über ein deutschlandweites

[11] http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html
[12] https://pangaea.de/
[13] http://orcid.org/

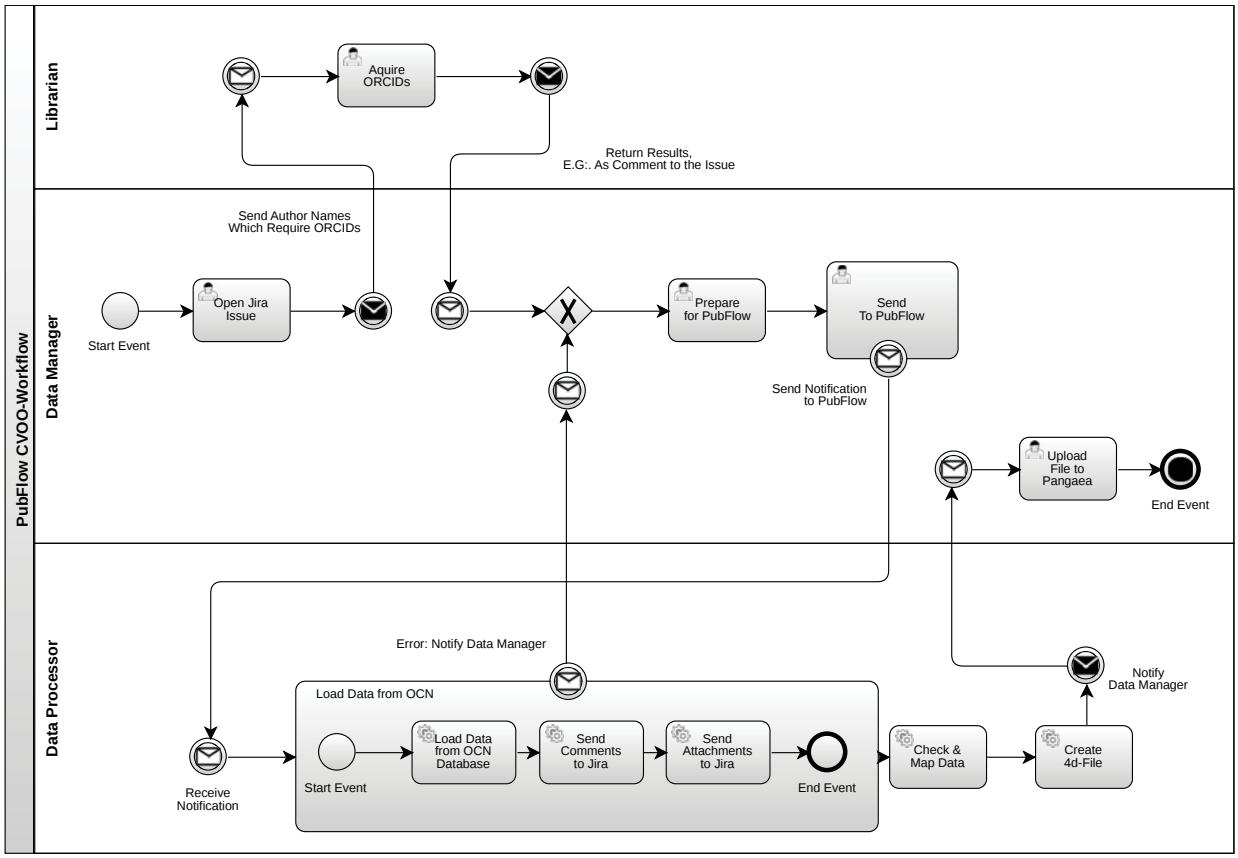

Abbildung 7: Workflow zur Veröffentlichung von Daten aus einer Datenbank (CVOO) in dem *World Data Center* PANGAEA

Konsortium ORCID beizutreten und die DNB arbeitet an der Verbindung von GND- und ORCID-Datensätzen. Pangaea entwickelt zurzeit eine neue Schnittstelle über die PubFlow automatisiert auf Autoreninformationen zugreifen und diese nachnutzen könnte. Diese Entwicklungen waren zum Zeitpunkt der Antragstellung nicht absehbar. Es war zwar möglich sie in den erstellten Konzepten zu berücksichtigen, eine Implementierung war hingegen ausgeschlossen.

Der Einsatz von Persistenten Identifikatoren zur eindeutigen Identifizierung von Personen ist von hoher Relevanz. Diese Aussage wird insbesondere mit Blick auf die Integration der ORCID durch die Fortschritte untermauert, die in der Zusammenarbeit von Pangaea mit dem Projekt Thor während der Projektlaufzeit von PubFlow erreicht wurden.[14] Das aktuell laufende DFG-Projekt ORCID-DE bestätigt zudem die Beschäftigung mit der Verbindung von ORCID und der im Bereich der Bibliotheken genutzten GND.[15] Im PubFlow-Projekt wurde die ORCID als bevorzugter Identifikator gewählt.

Es wird die Rolle des ‘Librarian’ in den Ablieferungsprozess integriert, wobei diese Rolle derzeit noch weitgehend auf die manuelle Mitwirkung von Bibliothekaren beschränkt ist. Der neu eingeführte ‘Librarian’ übernimmt in Zukunft die Aufgabe Personennamen eindeutig ORCID-Identifiern zuzuordnen. Abhängig von den Vereinbarungen innerhalb einer Einrichtung prüft und pflegt er dabei die Profildaten der Angehörigen der eigenen Einrichtung. Autoren ohne ORCID werden um die Erstellung einer ORCID gebeten und dabei unterstützt.

Während der auf ein Jahr beschränkten zweiten Projektphase wurden die Arbeitsabläufe in PubFlow mit besonderem Augenmerk auf die Integration der Personenidentifikation und deren Nutzung beim Management sowie der Veröffentlichung der Forschungsdaten analysiert und mo-

[14] vgl. ORCID-Integration, `https://project-thor.eu/2016/08/10/orcid-integration-in-pangaea/` (Stand 27.09.2017)

[15] vgl. ORCID-DE Projekt, `http://www.orcid-de.org/projekt/` (Stand 27.09.2017)

delliert. Innerhalb der kurzen Projektlaufzeit war es nicht möglich, diese Ergebnisse im PubFlow System zu implementieren. Die Ergebnisse der Analyse und die darauf aufbauenden Konzepte werden im Folgenden vorgestellt.

5.1 Auswahl des Identifikationssystems

Beim Entwurf des Projektantrags für die zweite Projektphase wurde ein Arbeitspaket zur Integration der Autorenidentifikation in das PubFlow System entwickelt, das auf die Verwendung der in der Gemeinsamen Normdatei (GND) enthaltenen Personennormsätze abzielte. Bis zur Aufnahme der Projektarbeit hatte mit Blick auf den Einsatz und die Verbreitung von ORCID in Deutschland eine dynamische Entwicklung stattgefunden. Ausdruck dafür waren das absehbar startende DFG-Projekt ORCID-DE und die auf Kieler Vorschlag geschaffene Möglichkeit Personen-IDs aus Fremdsystemen in der GND zur besseren Verknüpfung von Daten nachzuweisen. Durch diese Entwicklungen angestoßen, wurde der Ansatz zum Einsatz der GND überprüft. Der exemplarische Anwendungsfall aus den Meereswissenschaften lieferte wichtige Hinweise, die in der Überprüfung die Tauglichkeit der GND als primäres System zur Autorenidentifikation ebenfalls in Frage stellten. So ist die GND weitgehend auf den deutschsprachigen Raum beschränkt und Autoren erhalten erst eine GND, wenn Sie bereits publiziert haben. Aktualisierungen der GND-Daten können nur durch Bibliothekare erfolgen, die häufig erst mit langer Verzögerung von Änderungen, wie einem Wechsel der institutionellen Zugehörigkeit, Kenntnis erlangen. Entscheidend für den Wechsel zur ORCID als primäre Personen-ID in Forschungsdatenprozessen waren insbesondere die internationale Verbreitung, die Erwartung einer rascheren Aktualisierung durch den Autor sowie deren Unabhängigkeit von der Existenz von Publikationen. Um eine nachträgliche Verknüpfung von Forschungsdaten und Publikationen zu erleichtern, wurde an der UB Kiel eine Konkordanz zwischen ORCID und GND erstellt. Eine kommentierte Veröffentlichung der Konkordanz ist derzeit in Vorbereitung.

5.2 Autorenidentifikation im Publikationsprozess von PubFlow

PubFlow erfüllt prototypisch wichtige Anforderungen einer halbautomatisierten Vorbereitung und Veröffentlichungen von Forschungsdaten in ein definiertes Zielrepositorium (PANGAEA). PubFlow ist keine durchgängig automatisierte Lösung, sondern integriert auch manuelle, teilweise außerhalb von PubFlow organisierte Prozesse. Ziel ist, die gemeinsame Publikation von Forschungsdaten und der persönlichen ORCID-Identifier beteiligter Autoren zu ermöglichen. So sind Publikationen und ihre Autoren eindeutig zuzuordnen und nach ihrer Veröffentlichung eindeutig identifizier- und verknüpfbar. Abbildung 8 zeigt die implementierte PubFlow-Lösung im Zusammenwirken von Prozessen, beteiligten Rollen und heterogenen Infrastruktur-Bestandteilen.

Die einzelnen Bestandteile derAbbildung 8 wurden in vorangegangen Abschnitten bereits erläutert und werden an dieser Stelle nicht im Einzelnen beschrieben. Neu hinzugekommen sind als Folge der Analyse und Modellierung in der zweiten Projektphase die Rollen des „co-authors“ und des „librarian“ sowie die Datenquelle „ORCID“.

5.3 Modelle zur Erfassung und Integration von Autoren-IDs

Die Rolle des „co-authors“, steht für alle Co-Autoren, die an der Datenpublikation beteiligt sind. Gilt die Anforderung alle im System erfassten Autoren mit einem eindeutigen Identifier zu erfassen, bedeutet der Anspruch, diese Verpflichtung auch für die Co-Autoren einzuhalten, für den abliefernden Autor und / oder den „librarian“ einen erheblichen Mehraufwand, da die ORCIDs in aller Regel erst ermittelt werden müssen. In der Konzeption wurde versucht, diesem Anspruch gerecht zu werden und im Hinblick auf den Aufwand akzeptable Lösungen zu entwerfen. Um die anfangs noch als möglich erachtete Referenzimplementierung nicht völlig auszuschließen, wurde ein Modell mit drei verschiedenen Entwicklungsstufen erarbeitet. In der ersten Stufe werden

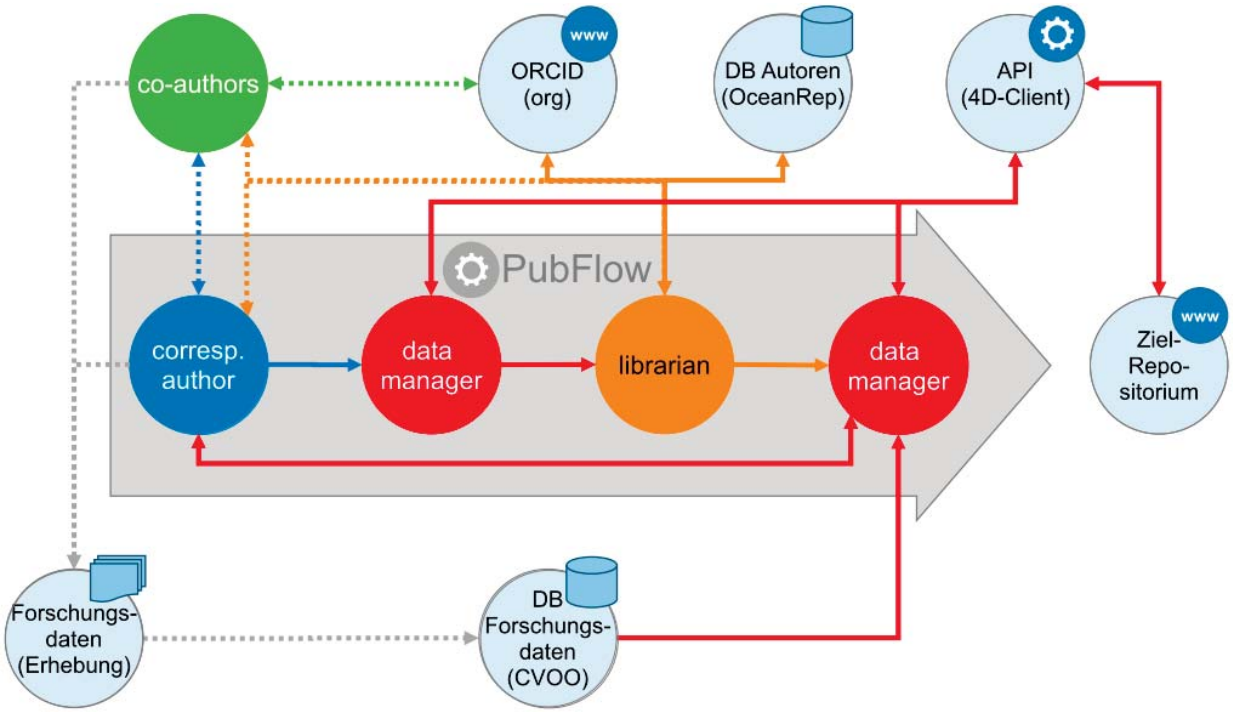

Abbildung 8: Die Publikationsprozesse in PubFlow und periphäre Anwendungen und Datenquellen

die Personen-IDs von Autoren vom abliefernden Autor, soweit bekannt, eingetragen und vom „librarian“ ergänzt, soweit ermittelbar (vgl. Abbildung 9).

Das zweite Modell sieht eine Autorenidentifikation über Schnittstellen vor (vgl. Abbildung 10). Dabei wird im Moment der Datenerfassung eine Suche über die ORCID-API durchgeführt, die anhand von Namen und E-Mailadressen Vorschlagslisten zur Auswahl von ORCIDs an den abliefernden Autor zurückliefert. In diesem Modell kann die Kommunikation mit den Co-Autoren voraussichtlich erheblich reduziert werden. Es ist außerdem zu erwarten, dass der Anteil von nachträglich durch den „librarian“ zu ermittelnden Personen-IDs s deutlich geringer ausfällt.

Beide bisher dargestellten Modellen erfordern die sich wiederholende Erfassung von Autoreninformationen, einschließlich ihrer Personen-IDs, bei jeder Ablieferung von Forschungsdaten. Vorteil dieses Ansatzes ist, dass, eine sorgfältige Ermittlung der Daten vorausgesetzt, aktuelle Personendaten erfasst werden. Nachteil ist, dass die Autoren bei der effizienten Ablieferung von qualitativ hochwertigen Metadaten, einer von ihnen häufig als unbeliebt angesehenen Tätigkeit, nicht optimal unterstützt werden. Um diesem Umstand Rechnung zu tragen, wurde ein weiterer Lösungsvorschlag entwickelt (vgl. Abbildung 11). In diesem Modell werden einmal erhobene und vom „librarian“ überprüfte und angereicherte Autoreninformationen in einer Datenbank gespeichert. Bei der zukünftigen Ablieferung von Forschungsdaten werden die vorhandenen Autorendaten dem abliefernden Autor über einen Suchindex angeboten und können so beispielsweise auch für Autovervollständigungsvorschläge genutzt werden. Die große Herausforderung in diesem Modell ist, dass die einmal erhobenen Daten veralten. So können insbesondere E-Mailadressen und institutionelle Zugehörigkeit („affiliation“) im wissenschaftlichen Umfeld in rascher Folge wechseln. Namensänderungen sind vergleichsweise seltener, kommen aber ebenfalls vor. Um die erforderliche Aktualität der Daten zu gewährleisten, sieht das Modell daher einen regelmäßigen Abgleich der Daten anhand der gespeicherten Personen-IDs mit dem Quellsystem (ORCID) vor. Der Abgleich erfolgt vollautomatisch über Batch-Prozesse.

Aufgrund der in der zweiten Projektphase verfügbaren Entwicklerkapazitäten und der Priorisierung der Projektaufgaben konnten die erweiterten Modelle nicht implementiert werden. Das aktuelle PubFlow System entspricht heute weitgehend dem in Abbildung 9 dargestellten Modell.

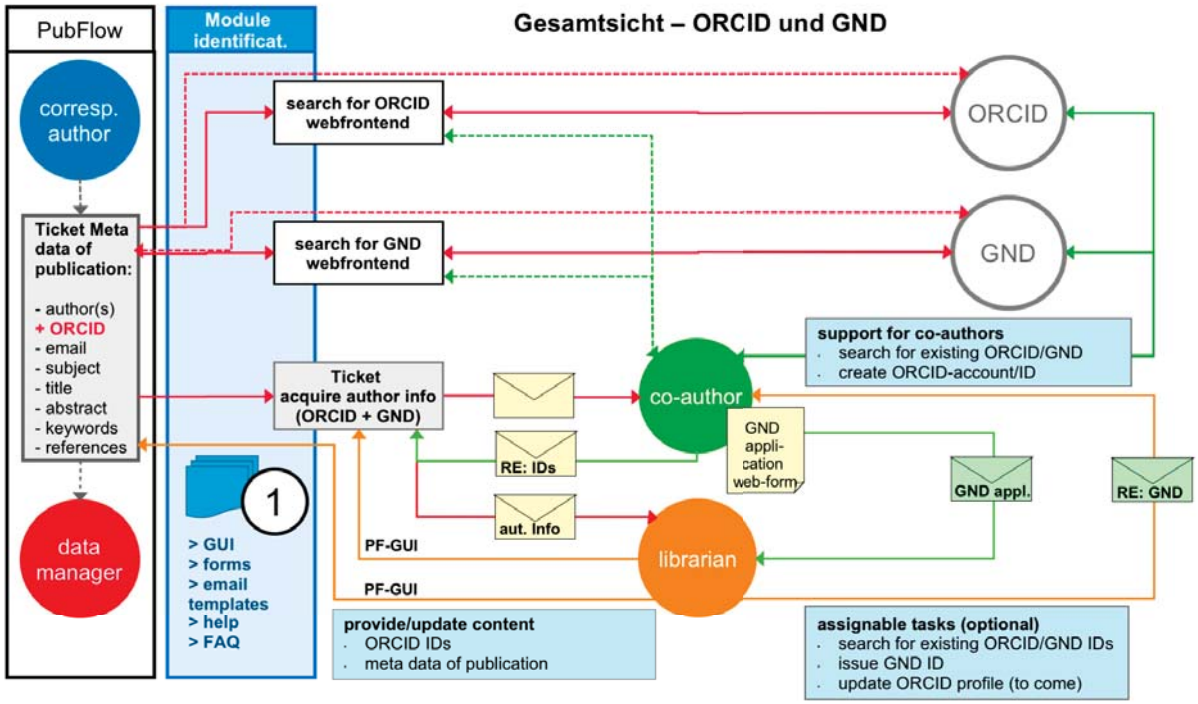

Abbildung 9: Schematischer Ablauf der Autorenidentifikation im PubFlow System, ohne Abfrage von Daten über technische Schnittstellen (API). Die Darstellung stellt im Vergleich zum gewählten Ansatz eine Verallgemeinerung dar, da auch die Erfassung von GNDs berücksichtigt ist.

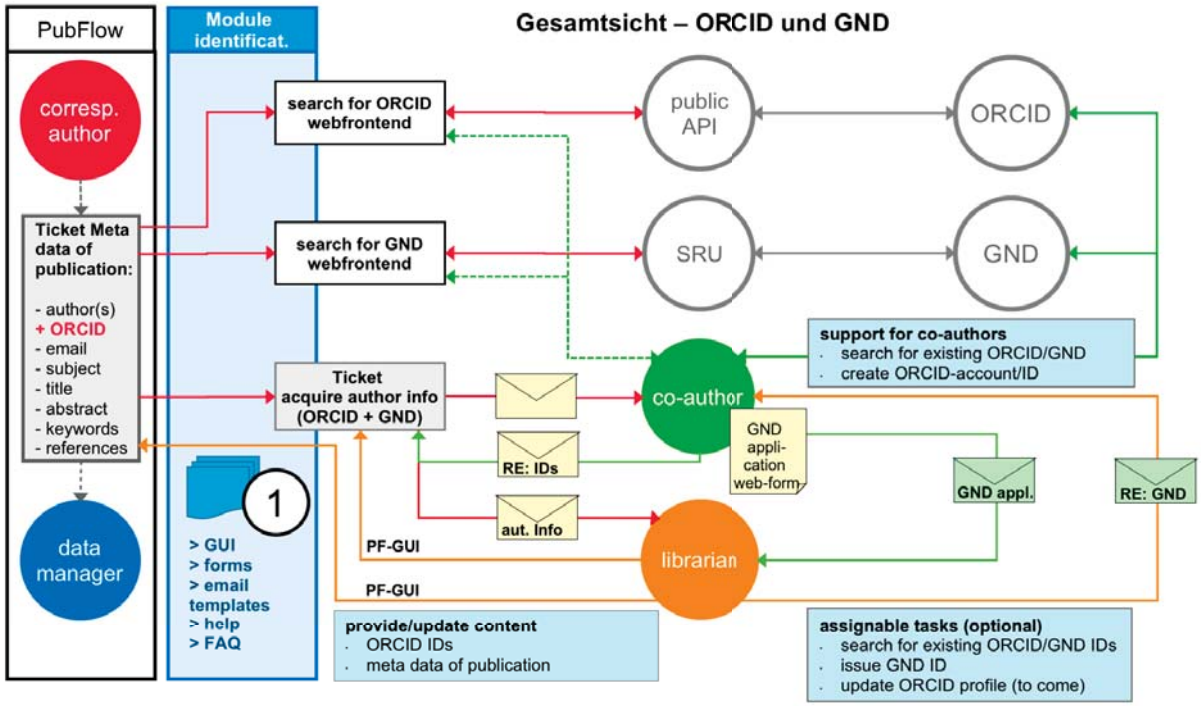

Abbildung 10: Schematischer Ablauf der Autorenidentifikation im PubFlow System unter Nutzung von Schnittstellen zur Abfrage von Personen-IDs. Die Darstellung stellt im Vergleich zum gewählten Ansatz eine Verallgemeinerung dar, da auch die Abfrage und Erfassung von GNDs berücksichtigt ist.

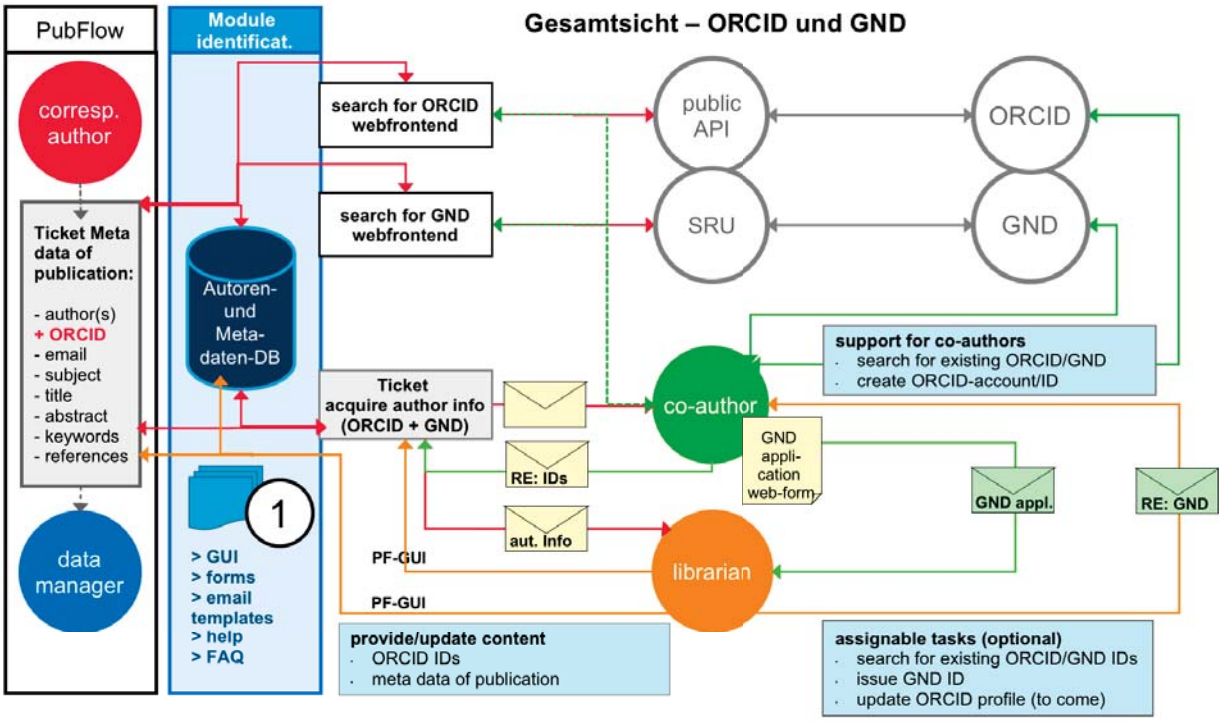

Abbildung 11: Schematischer Ablauf der Autorenidentifikation im PubFlow System unter Nutzung von Schnittstellen zur Abfrage von Personen-IDs. Als zusätzliche Komponenten enthält dieser Lösungsvorschlag eine Datenbank zur Zwischenspeicherung von bereits bekannten Autoreninformationen. Die Darstellung stellt im Vergleich zum gewählten Ansatz eine Verallgemeinerung dar, da auch die Abfrage und Erfassung von GNDs berücksichtigt ist.

6 Erfahrungen mit den eingesetzten Systemen

Das im Rahmen des Projektes genutzte Ticketsystem wurde gut von den Nutzerinnen und Nutzern angenommen. Dadurch, dass jeder Vorgang durch ein einzelnes Ticket repräsentiert wird, sind alle relevanten Informationen an einem Ort gebündelt und der Verlauf und potentielle Fehler können leicht nachvollzogen werden. Zusätzlich ist der Ablauf klar definiert und kann jederzeit überprüft werden. Bei Jira ist es einfach möglich, neue Workflows hinzuzufügen. Allerdings ist es in unserem Prototypen für die normalen Nutzer und Nutzerinnen nicht möglich neue Workflows anzulegen, die andere Services und Funktionen nutzen, als die von Jira schon vorgegebenen.

In der ersten Version des Prototypen wurde die Kommunikation über einen Nachrichtenbus, bestehend aus ActiveMQ[16] und Apache Camel[17], realisiert. Diese wurden später entfernt, da die Komplexität des Systems den Nutzen negiert hat. In dieser ersten Version war weiterhin die Zielsetzung die Workflows von den Datenmanagerinnen und Datenmanagern in BPMN generieren zu lassen und diese dann in ausführbare BPEL-Modelle zu übersetzen [SH10]. Dieses Feature wurde bei der Umstellung auf Jira nicht mit migriert. Bei der Umstellung wurden für die manuellen Schritte der Workflows, die von den Nutzerinnen und Nutzern umgesetzt werden, Jira-Workflows modelliert. Zusätzliche wurde ein großer Teil der vorher bestehenden Software in ein Plugin für Jira migriert. Außerdem wurden Webservices für die Verbindung zu den lokalen institutionellen Datenbanken angelegt. Die von PubFlow automatisierten Teile der Workflows werden weiterhin in BPMN modelliert. Dabei werden die einzelnen Schritte per Hand um Java-Code angereichert und dann über die BPMN-Engine *jBPM* von jBoss ausgeführt. Das führt zum einen dazu, dass für neue Workflows in vielen Fällen neuer Java-Code erzeugt werden muss. Zum anderen existieren ab diesem Zeitpunkt zwei verschiedene Komponenten die Workflows ausführen.

Die Entscheidung, den größten Teil des PubFlow-Prototypen als Jira-Plugin zu implementieren, führte zu starken Abhängigkeiten zu den Updatezyklen von JIRA und dessen Schnittstellen. Dies führte zu komplexen Anpassungsarbeiten um den Prototypen von der älteren Jira-Version für die Nutzung von Jira 7 anzupassen.

Durch die Modernisierung der Architektur des PubFlow-Prototypen wurde die Wartung der Software spürbar einfacher. Sobald ein neuer Service benötigt wird, kann dieser einfach implementiert werden, ohne andere Komponenten zwangsläufig zu verändern. Allerdings müssen verschiedene Elemente des Systems angefasst werden, sobald ein neuer, komplexer Workflow eingeführt werden soll. Ein weiterer inhärenter Vorteil der Microservice-Architektur ist es, dass wir verschiedene Technologien für verschiedene Services wählen können. Dies birgt auch das Potential für die vereinfachte Zusammenarbeit mit den Kooperationspartnern. So ist es auch einfach möglich, Schnittstellen zu definieren und Services von verschiedenen Partnern erstellen zu lassen. Dadurch konnte zum Beispiel der Datenimport in die Datenbank des GEOMAR von einem Webservice übernommen werden, der vom dortigen Datenmanagementteam erstellt wurde.

Die Aufspaltung des Monolithen in mehrere Microservices erwies sich als aufwändig. Besser wäre es gewesen die Architektur von vorn herein als Microservice-Architektur zu planen und mit einzelnen Services zu starten [HS17].

Eine Herausforderung verteilter Systeme ist es das System und die übermittelten Informationen abzusichern. Da die unterschiedlichen Komponenten auf verschiedenen Rechnern verteilt sein können, müssen auch klare Regeln für die Zugriffsberechtigung der einzelnen Serviceschnittstellen geschaffen werden. Des weiteren sollen auch die versendeten Nachrichten nicht für Dritte lesbar sein. In unserem Prototypen benutzen wir deshalb eine Kombination aus Portregeln und OAuth[18]. Dabei sind alle Komponenten, bis auf Jira, nur innerhalb des Netzwerkes aufrufbar und die Schnittstelle von Jira (bzw. dss PubFlow Plugins) kann nur von autorisierten Programmen genutzt werden.

[16] http://activemq.apache.org/
[17] http://camel.apache.org/
[18] https://de.wikipedia.org/wiki/OAuth

7 Zusammenfassung

In der ersten Projektphase von PubFlow wurde zunächst ein Konzept für Publikationsprozesse für Forschungsdaten geschaffen und eine Pilotanwendung für die Meereswissenschaften am Standort Kiel umgesetzt. Dies wurde mit dem Ziel umgesetzt eine flexible und verlässliche Plattform zu bieten, mit der die Forschungsdaten primär für die eigene Organisation verwaltet werden. Die Langzeitarchivierung ist über eine Anbindung des World Data Center for Marine Environmental Sciences (WDC-MARE) ermöglicht, welches über PANGAEA zugänglich ist. Zu Beginn des Projektes hat es sich in Gesprächen als sehr wichtig herausgestellt, die Komplexität der Workflows vor den Nutzern des Systems zu verbergen. Es wurde daher entschieden verschiedenen Nutzerrollen (z.B. Datenmanager, Wissenschaftler der Fachdomäne, ...) verschiedene Sichten auf das System zu zuordnen. Wissenschaftlern der Fachdomäne wurden die Funktionen des PubFlow-Systems in Form eines Plugins angeboten. Datenmanager hingegen konnten dort neue Publikationsworkflows anlegen und bestehende Workflows bearbeiten. Weiterhin wurden verschiedene Workflow-Plattformen evaluiert, auf denen das PubFlow-System realisiert werden sollte. Die erste prototypische Realisierung des PubFlow-Frameworks wurde in den ersten Projektjahren erfolgreich realisiert und bereits durch das Datenmanagementteam am GEOMAR evaluiert. In der darauf folgenden Version wurde die Nutzeroberfläche durch die Einbindung des Ticketsystems Jira ersetzt um die Akzeptanz des Tools zu fördern.

In der zweiten Projektphase des PubFlow Projektes konnten wir, aufbauend auf der ersten Projektphase, die Architektur unseres Prototypen modernisieren und einen ersten Ansatz zur Autorenidentifikation während der Publikation von wissenschaftlichen Daten umsetzen. Die Änderung der Architektur von einer klassischen Schichtenarchitektur hin zu einer Microservice-Architektur erhöht insbesondere die Wartbarkeit, als auch die Möglichkeit das System partiell zu erweitern und anzupassen. Dadurch ist es weiterhin einfacher für andere Parteien den Prototypen, als ein Ergebnis des Projektes, weiter zu nutzen und selbst weiter zu entwickeln. Gegebenenfalls können so auch nur einzelne Systemteile oder Services für andere Kontexte genutzt werden. Erste Ansätze zur automatisierten, workflowbasierten Provenienzdatenerhebung wurden erarbeitet.

In umfangreichen Diskussionen mit den Projektpartnern, Anwendern und Bibliotheken konnten wir Ansätze zur Autorenidentifizierung erarbeiten. Insbesondere soll es einer Organisation möglich sein Autoren und Autorinnen aus dem eigenen, lokalen Datenbeständen mit globalen, eindeutigen Identifikatoren zu verknüpfen. Da es durchaus vorkommen kann, dass im Rahmen von Kooperationen Beteiligte aus anderen Organisationen mit beachtet werden müssen, spielen verschiedene Faktoren eine Rolle: Beispielsweise die Wahl des Identifikators, der Anspruch auf Nachnutzbarkeit bereits ermittelter IDs, mögliche ID-Dopplungen, Namensgleichheiten und -änderungen, Aktualität der eigenen Daten, etc. In unserem Prototypen konnten wir einen ersten Entwurf zur Identifizierung umsetzen. Dabei wird insbesondere die Rolle des *Librarian* eingeführt, die mit nötigem Fachwissen die Identifizierung vornehmen soll. Das entwickelte Konzept zur Integration der Autorenidentifikation basiert im Kern auf einer umfassenden Analyse der bestehenden Prozesse und einem Verständnis der Rollen, Prozesse und Anforderungen das PubFlow-System. Das Konzept ist mehrstufig angelegt, so dass abhängig von den Implementierungsschritten verschiedene Integrations- und Automatisierungsgrade erreicht werden können.

Es gibt verschiedene Punkte in denen das in diesem Projekt entwickelte Vorgehen und der Prototyp verbessert und weiterentwickelt werden können. Es ist beabsichtigt, die entwickelten Konzepte zur Integration der Personenidentifikation in die Forschungsdatenmanagement-Workflows zu veröffentlichen. Die Konzepte fließen in den Aufbau und die Erweiterung von bibliothekarischen Dienstleistungen zur Erstellung und Pflege von Personenprofilen und Personennormdaten mit ein. Weiterhin gibt es Überlegungen PubFlow auch im Projekt GeRDI zu nutzen [Gru+17].

Die Homepage zum PubFlow-Projekt ist unter `http://www.pubflow.uni-kiel.de` zu finden. Für Testzwecke stellen wir unter `http://maui.se.informatik.uni-kiel.de:48080/jira` eine Live-Demo zur Verfügung. Das Jira-Plugin und alle im Projekt erstellten Services sind unter `https://github.com/pubflow` zu finden. Die Software steht unter der *Apache 2.0 License*.

Literatur

[AH16] Marc Adolf und Wilhelm Hasselbring. „Einsatz kommerzieller und Open-Source Software für wissenschaftliche Workflows zur Datenpublikation in PubFlow“. In: *5. DINI / nestor-Workshop Werkzeuge für Forschungsdaten: Bedarf und Integration in Forschungs- und Datenmanagementprozesse*. Juni 2016. URL: http://eprints.uni-kiel.de/33149/.

[AW08] Richard Anderson und B.D. McCullough und H.D. Vinod William H. Greene. „The role of data & program code archives in the future of economic research“. In: *Journal of Economic Methodology* 15 (2008), S. 99–115.

[Bra12a] Peer Christoph Brauer. „Capturing provenance information with Kieker.Workflow-Monitor“. In: *1st ZBW International PhD Summer School*. Juni 2012. URL: http://eprints.uni-kiel.de/23255/.

[Bra12b] Peer Christoph Brauer. „Kieker.Workflow - Workflow Monitoring mit Kieker“. In: *KoSSE-Symposium Application Performance Management (Kieker Days 2012)?* Nov. 2012. URL: http://eprints.uni-kiel.de/19635/.

[Bra12c] Peer Christoph Brauer. „Provenance data archival in PubFlow“. In: *SWIB12*. Nov. 2012. URL: http://eprints.uni-kiel.de/20551/.

[Bra14] Peer Christoph Brauer. „CAPS: Capturing and Managing Provenance Information in Scientific Workflows“. In: *ZBW PHD Springschool 2014*. ZBW, März 2014. URL: http://eprints.uni-kiel.de/23929/.

[BCH14] Peer Christoph Brauer, Andreas Czerniak und Wilhelm Hasselbring. „Start Smart and Finish Wise: The Kiel Marine Science Provenance-Aware Data Management Approach“. In: *6th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2014)*. Cologne, Germnay: USENIX Association, Juni 2014. URL: http://eprints.uni-kiel.de/24972/.

[BFH14] Peer Christoph Brauer, Florian Fittkau und Wilhelm Hasselbring. „The Aspect-Oriented Architecture of the CAPS Framework for Capturing, Analyzing and Archiving Provenance Data“. In: *5th International Provenance and Annotation Workshop (IPAW 2014)*. Lecture Notes in Computer Science. Springer-Verlag, 2014. URL: http://eprints.uni-kiel.de/24726/.

[BH12] Peer Christoph Brauer und Wilhelm Hasselbring. „Capturing provenance information with a workflow monitoring extension for the Kieker framework“. In: *Proceedings of the 3rd International Workshop on Semantic Web in Provenance Management*. Bd. 856. CEUR Workshop Proceedings. CEUR-WS, Mai 2012. URL: http://eprints.uni-kiel.de/19636/.

[BH13a] Peer Christoph Brauer und Wilhelm Hasselbring. „PubFlow: a scientific data publication framework for marine science“. In: *Proceedings of the International Conference on Marine Data and Information Systems (IMDIS 2013)*. Bd. 54. Sep. 2013, S. 29–31. URL: http://eprints.uni-kiel.de/22399/.

[BH13b] Peer Christoph Brauer und Wilhelm Hasselbring. „PubFlow: provenance-aware workflows for research data publication“. In: *5th USENIX Workshop on the Theory and Practice of Provenance (TaPP '13)*. Apr. 2013. URL: http://eprints.uni-kiel.de/21112/.

[Con+05] Stefan Conrad u. a. *Enterprise Application Integration*. Spektrum Akademischer Verlag, 2005.

[DTA86] William G. Dewald, Jerry G. Thursby und Richard G. Anderson. „Replication in Empirical Economics: The Journal of Money, Credit and Banking Project“. In: *American Economic Review* 76.4 (1986), S. 587–603.

[Gru+17] Richard Grunzke u. a. „Challenges in Creating a Sustainable Generic Research Data Infrastructure“. In: *4th Collaborative Workshop on Evolution and Maintenance of Long-Living Software Systems*. Softwaretechnik-Trends, Feb. 2017.

[Gud+08a] Stefan Gudenkauf u. a. „A Software Architecture for Grid Utilisation in Business Workflows“. In: *Multikonferenz Wirtschaftsinformatik 2008 (MKWI 2008)*. Hrsg. von Martin Bichler u. a. GITO-Verlag, Berlin, Feb. 2008. ISBN: 978-3-940019-34-9.

[Gud+08b] Stefan Gudenkauf u. a. „BIS-Grid: Business Workflows for the Grid“. In: *Proc. Cracow Grid Workshop 2007 (CGW'07)*. Hrsg. von Marian Bubak, Michal Turala und Kazimierz Wiatr. Krakow, Poland: ACC CYFRONET AGH, 2008, S. 86–94. ISBN: 978-83-915141-9-1.

[Gud+08c] Stefan Gudenkauf u. a. „Workflow Service Extensions for UNICORE 6 – Utilising a Standard WS-BPEL Engine for Grid Service Orchestration“. In: *Euro-Par 2008 Workshops – Parallel Processing*. Bd. 5415. Lecture Notes in Computer Science. Las Palmas de Gran Canaria, Spain: Springer, Aug. 2008, S. 103–112. DOI: 10.1007/978-3-642-00955-6_13.

[Gud+10] Stefan Gudenkauf u. a. „Workflow Modeling for WS-BPEL-based Service Orchestration in SMEs“. In: *Software Engineering 2010 – Workshopband*. Hrsg. von Gregor Engels u. a. Bd. 160. LNI. GI, 2010, S. 185–192. ISBN: 978-3-88579-254-3.

[Has00] Wilhelm Hasselbring. „Information System Integration“. In: *Communications of the ACM* 43.6 (2000), S. 32–36.

[Has02] Wilhelm Hasselbring. „Component-Based Software Engineering“. In: *Handbook of Software Engineering and Knowledge Engineering*. World Scientific Publishing, 2002, S. 289–305.

[Has06] Wilhelm Hasselbring. „Software-Architektur – Das aktuelle Schlagwort“. In: *Informatik-Spektrum* 29.1 (Feb. 2006), S. 48–52.

[Has09] Wilhelm Hasselbring. *Wisent: Wissensnetz Energiemeteorologie: Schlussbericht*. GITO mbH Verlag, 2009.

[Has10] Wilhelm Hasselbring, Hrsg. *Betriebliche Informationssysteme: Grid-basierte Integration und Orchestrierung*. GITO-Verlag, 2010. ISBN: 978-3-942183-20-8.

[Has16] Wilhelm Hasselbring. „Microservices for Scalability: Keynote Talk Abstract“. In: *Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering (ICPE 2016)*. Delft, The Netherlands: ACM, 2016, S. 133–134. ISBN: 978-1-4503-4080-9. DOI: 10.1145/2851553.2858659.

[HS17] Wilhelm Hasselbring und Guido Steinacker. „Microservice Architectures for Scalability, Agility and Reliability in E-Commerce“. In: *Proceedings 2017 IEEE International Conference on Software Architecture Workshops (ICSA 2017)*. Gothenburg, Sweden: IEEE, Apr. 2017.

[Has+06] Wilhelm Hasselbring u. a. „WISENT: e-Science for Energy Meteorology“. In: (Dez. 2006), S. 93–100.

[Kel+15] C. Brenhin Keller u. a. „Volcanic-plutonic parity and the differentiation of the continental crust“. In: *Nature* 523.7560 (2015). Article, S. 301–307. URL: http://dx.doi.org/10.1038/nature14584.

[New15] Sam Newman. *Building Microservices*. O'Reilly Media, Inc., 2015.

[PDF07] Heather A. Piwowar, Roger S. Day und Douglas B. Fridsma. „Sharing Detailed Research Data Is Associated with Increased Citation Rate“. In: *PLOS ONE* 3 (2007), S. 1–5. DOI: 10.1371/journal.pone.0000308. URL: https://doi.org/10.1371/journal.pone.0000308.

[Plo+09] Jan Ploski u. a. „Grid-based deployment and performance measurement of the Weather Research Forecasting model“. In: *Future Generation Computer Systems* 25.3 (2009), S. 346–350. DOI: http://doi.org/10.1016/j.future.2008.05.003. URL: http://www.sciencedirect.com/science/article/pii/S0167739X08000605.

[SH10] Guido Scherp und Wilhelm Hasselbring. „Towards a Model-Driven Transformation Framework for Scientific Workflows“. In: *International Conference on Computational Science (ICCS 2010)*. Procedia Computer Science. 2010, S. 1513–1520. DOI: DOI: 10.1016/j.procs.2010.04.169.

[SH11] Guido Scherp und Wilhelm Hasselbring. „Interoperability of the BIS-Grid Workflow Engine with Globus Toolkit 4“. In: *Proceedings of the Grid Workflow Workshop 2011*. Hrsg. von Klaus-Dieter Warzecha und Lars Packschies. Bd. 826. CEUR Workshop Proceedings, 2011. URL: http://ceur-ws.org/Vol-826/.

[Sch+10] Guido Scherp u. a. „Using UNICORE and WS-BPEL for Scientific Workflow Execution in Grid Environments“. In: *Proc. EuroPar 2009 Parallel Processing Workshops*. Hrsg. von H.-X. Lin u. a. Bd. 6043. Lecture Notes in Computer Science. Springer, 2010, S. 335–344. ISBN: 978-3-642-14121-8. DOI: 10.1007/978-3-642-14122-5.

[Sci11] Science. *Data Replication & Reproducibility*. http://www.sciencemag.org/site/special/data-rep/. 2011.