

---

# Lokal betrieben, remote gepflegt – Software für ein Datenrepositorium in Kooperation implementieren

Matthias Landwehr <sup>1</sup>, Gabriel Schneider <sup>1</sup>, Stefan Hofmann <sup>2</sup>, Matthias Razum <sup>2</sup> und Kerstin Soltau <sup>2</sup>

<sup>1</sup>Kommunikations-, Informations-, Medienzentrum (KIM), Universität Konstanz

<sup>2</sup>FIZ Karlsruhe – Leibniz-Institut für Informationsinfrastruktur

Die Universität Konstanz deckt ihren Bedarf nach einem institutionellen Forschungsdatenrepositorium mit der von FIZ Karlsruhe angebotenen Lösung „RADAR Local“. Als Alternative zu einer Eigenentwicklung wurde das Datenrepositorium als hybrides Modell mit Repositorien-Software und Archivierung auf lokaler Infrastruktur implementiert. Dabei stellt FIZ Karlsruhe die etablierte Repositorien-Software RADAR zur Verfügung, wartet und betreibt sie aus der Ferne und passt sie nach Kundenwunsch an. Um die parallele Installation und Pflege der RADAR-Software auf mehreren lokalen Instanzen effizient bewältigen zu können, hat FIZ Karlsruhe vorab den Automatisierungsgrad der betroffenen Prozesse in der Software-Entwicklung, in der Systemkonfiguration und im Deployment erhöht. Dies wurde durch den Einsatz von Container-Virtualisierung wie Docker und Docker Swarm sowie mit Orchestrierungswerkzeugen wie Ansible erreicht.

Der zeitliche Aufwand und der personelle Ressourcenbedarf reduzieren sich dadurch für die Universität Konstanz und als Ergebnis erhält sie ein gepflegtes Repositorium auf dem aktuellen Stand der Technik. Gleichzeitig erfordert diese Betriebsvariante eine intensive Auseinandersetzung mit dem jeweiligen Geschäftsmodell und den technischen Rahmenbedingungen des Anbieters, eine genaue Kostenkalkulation sowie möglicherweise Kompromisse oder Abstriche bei individuellen Wünschen.

## 1 Einleitung

Universitäten und andere wissenschaftliche Einrichtungen sehen sich bei der Entwicklung und Einführung von Datenrepositorien vielschichtigen Herausforderungen gegenüber. Am Anfang steht der Bedarf nach einem institutionellen Repositorium, die Herausforderung liegt in der Umsetzung.

Neben der Wahl einer geeigneten technischen Basis ist der Aufwand einer individuellen Implementierung und langfristigen Pflege nicht zu unterschätzen.

Erschwerend hinzu kommt der aktuelle IT-Fachkräftemangel, der sich gerade in öffentlichen Einrichtungen zunehmend niederschlägt und zu großen Verzögerungen bei Eigenentwicklungen im IT-Bereich führt. Diesen Problemen sieht sich auch die Universität Konstanz gegenüber.

## 2 Bedarf

Am Ausgangspunkt der Kooperation stand der Bedarf der Universität Konstanz nach einem eigenen institutionellen Datenrepositorium. Dieser Bedarf bildete sich in den vergangenen Jahren aus verschiedenen Gründen heraus. Die Bedeutung von Forschungsdaten und deren Management hat in den letzten Jahren stetig zugenommen. Nicht zuletzt durch die Richtlinien von nationalen und internationalen Forschungsförderern, ist die eigenständige Publikation von Forschungsdaten in zitierfähiger Form eine Aufgabe geworden, mit der sich Forschende beschäftigen müssen und auch wollen. Zusätzlich stellt das Thema Open Science einen Schwerpunkt der Universität Konstanz dar und das Anbieten eines institutionellen Datenrepositoriums kann einen Beitrag zur Öffnung der Wissenschaft darstellen.

Die Universität Konstanz wurde im Jahr 1966 als Reformuniversität gegründet und ist seit 2007 dauerhaft in den Exzellenzwettbewerben des Bundes erfolgreich. Das Kommunikations-, Informations-, Medienzentrum (KIM) ist der Zusammenschluss von Bibliothek und Rechenzentrum und betreibt und pflegt die IT-Angebote für die Universität. Das Thema Open Science ist in der Exzellenzstrategie prominent verankert und das KIM unterstützt die Ziele mit einem gut aufgestellten Team Open Science, das sich u.a. um die Themen Open Access und Forschungsdatenmanagement kümmert.

### 2.1 Ausgangslage

Die erste Empfehlung für die Datenpublikation ist immer die Nutzung eines fachspezifischen Repositoriums. Leider findet sich in manchen Fällen kein geeigneter Dienst und dann sieht man sich als Universität und Infrastruktureinrichtung mit der Forderung konfrontiert, eine dauerhafte Heimat für diese Daten bereitzustellen. Repositorien können aufgrund von Veröffentlichungskosten, mangelnder Einschlägigkeit, Limitierungen in Bezug auf die Größe der Datensätze oder anderen Hürden in den Richtlinien unpassend für die Bedarfe der Wissenschaftler\*innen sein. In einigen Fällen ist es auch gewünscht, dass die publizierten Daten „physisch“ im Haus bleiben. Dann fallen internationale Repositorien mit außereuropäischen Servern z.B. aus Gründen des Datenschutzes heraus. Zudem hat sich auch das Selbstverständnis vieler Einrichtungen dahingehend verändert, dass ein institutionelles Datenrepositorium inzwischen zum Serviceangebot gehört.

Bei jedem Dienst, den man als Universität anbieten möchte, muss man die Ressourcensituation betrachten. Während technische und finanzielle Ressourcen das geringere Problem darstellen, steht das Thema der personellen Ressourcen im Vordergrund.

Der Fachkräftemangel schlägt - insbesondere in der IT - auch an Universitäten durch. In Konstanz kommt als zusätzlicher Faktor noch die Nähe zur Schweiz hinzu, die den Jobmarkt nochmal verengt.

Ebenfalls zu berücksichtigen ist die Tatsache, dass es bereits zahlreiche institutionelle Datenrepositorien gibt und es nicht angebracht ist, dass überall „das Rad neu erfinden“ werden muss. Im akademischen Umfeld gab es in der Vergangenheit viele Fälle von Einrichtungen mit nach einigen Jahren schlecht wartbarer Individualsoftware, die nicht nachhaltig gepflegt und betrieben werden konnte. Aus dieser Ausgangslage entwickelte sich die Anforderung, eine geeignete Lösung mit ausreichender Funktionalität zu finden, die mit geringem Aufwand zu realisieren ist.

## 2.2 Anforderungen

Der erste Schritt ist immer die Aufstellung der funktionalen Anforderungen, die ein Produkt erfüllen muss. In gemeinsamer Arbeit mit allen beteiligten Personen in der Universität wurde ein Kriterienkatalog erstellt. Dabei kooperierten vor allem Mitarbeiter\*innen des Bereichs Open Science und Software-Entwickler\*innen des KIM mit den verantwortlichen Personen aus der technischen Infrastruktur. Ebenfalls einbezogen wurde die bibliothekarische Kompetenz in den Bereichen Metadaten und Publikationen. Die wichtigsten Punkte des Katalogs werden hier dargestellt. Das gewünschte Datenrepositorium soll:

- ein generisches Repository für alle Arten von Datensätzen sein
- über ein ausreichendes Rollen- und Rechteverwaltung verfügen
- eine flexible Metadatenverwaltung mitbringen
- bei der Publikation einen DOI vergeben
- verschiedene Lizenzen anbieten, Embargo und Closed Access unterstützen
- intuitiv benutzbar sein
- sich in die bestehende IT-Landschaft der Universität integrieren lassen
- die Authentifizierung, Autorisierung und die Bereitstellung der Datensätze über etablierte Schnittstellen ermöglichen
- leicht wartbar sein und aktiv weiterentwickelt werden

Bevor im nächsten Schritt verschiedene Systeme grundlegend gegen diesen Kriterienkatalog evaluiert wurden, wurde die Frage gestellt, ob man ein Repository nicht als „Software / Infrastructure as a service“ komplett einkaufen kann. Bei der Bereitstellung eines Repositoriums durch einen externen Dienstleister in einer Cloud werden Wartung, Pflege und Entwicklung komplett abgetreten, der Kunde nutzt ein fertiges Produkt. Dies ermöglicht den Betrieb mit einem wesentlich geringeren Personalaufwand.

Statt Ressourcen für Serveradministration und Softwareinstallation zu binden, kann eine Konzentration auf die bibliothekarische Kernkompetenz stattfinden und den Nutzenden eine umfangreiche Unterstützung bei der Kuration und der Publikation der Daten zur Verfügung gestellt werden.

Ein Hindernis bei einer Cloud-Lösung ist die Tatsache, dass die Daten in der Cloud liegen und damit die lokale Infrastruktur der Universität verlassen. Bei der Recherche und Marktanalyse stieß die Universität Konstanz dann auf das Produkt „RADAR Local“ von FIZ Karlsruhe. Das Leibniz-Institut für Informationsinfrastruktur ist eine GmbH mit anerkannter Gemeinnützigkeit und hat als eine der großen Informationsinfrastruktureinrichtungen in Deutschland den öffentlichen Auftrag, Wissenschaft und Forschung mit wissenschaftlicher Information zu versorgen und entsprechende Produkte und Dienstleistungen zu entwickeln.

Mit RADAR bietet FIZ Karlsruhe seit 2017 Forscherinnen und Forschern an Hochschulen und außeruniversitären Forschungseinrichtungen in Deutschland ein flexibles, kostengünstiges und einfach zu nutzendes Repository für die Archivierung und Publikation von Forschungsdaten. Gemeinsam mit den nutzenden Einrichtungen wird RADAR kontinuierlich weiterentwickelt und an neue Nutzungskontexte angepasst. Mit der Betriebsvariante RADAR Local wurde ein Produkt geschaffen, das auf der Infrastruktur des Kunden läuft, aber von extern betrieben und gepflegt wird. RADAR Local ging in der Gesamtbetrachtung als Sieger aus der Evaluation verschiedener Produkte hervor.

## 3 Technische Umsetzung

### 3.1 Motivation

Das Forschungsdatenrepositorium RADAR<sup>1</sup> entstand als Cloud-Lösung im Rahmen eines DFG-Projekts<sup>2</sup> und ging 2017 online. Mittlerweile setzen zehn Hochschulen und außeruniversitäre Forschungseinrichtungen „RADAR Cloud“ ein. Die Einführung und Nutzung von RADAR Cloud ist für diese Institutionen sehr einfach, allerdings fallen volumenabhängige Speicherkosten an.

Auch gibt es vertragliche Beschränkungen bezüglich der Archivierung personenbezogener Forschungsdaten. In Gesprächen mit interessierten Institutionen entstand so die Idee, RADAR in einer weiteren Betriebsform anzubieten: die Einrichtung stellt die notwendige IT-Infrastruktur für den Betrieb des Systems, FIZ Karlsruhe stellt die Software und betreibt RADAR auf den lokalen Systemen der Einrichtung.

Aktuell läuft die RADAR-Software - eine verteilte, mehrschichtige Anwendung, die sich in eine Vielzahl von Diensten und Schnittstellen gliedert (siehe Abbildung 1) - nur auf

---

<sup>1</sup><https://www.radar-service.eu>

<sup>2</sup>[https://radar.products.fiz-karlsruhe.de/sites/default/files/radar/docs/info/Abschlussbericht\\_RADAR\\_DFG-Projekt\\_Publikation.pdf](https://radar.products.fiz-karlsruhe.de/sites/default/files/radar/docs/info/Abschlussbericht_RADAR_DFG-Projekt_Publikation.pdf)

einer zentralen Cloud-Instanz. Für RADAR Local kommen weitere Instanzen hinzu. Alle Instanzen nutzen dabei den identischen Software-Stack. Damit stellt sich die grundsätzliche Frage, wie FIZ Karlsruhe das komplexe System nicht nur effizient für RADAR Cloud installieren kann, sondern zukünftig parallel auch auf den Systemen aller Einrichtungen, die RADAR Local nutzen.

Die beiden Betriebsvarianten unterscheiden sich hauptsächlich in der Storage Layer, denn hier findet die Verknüpfung mit dem institutseigenen Archivspeicher statt. Um den Aufwand für Installation und Pflege möglichst gering zu halten, war das Ziel eine weitestgehende Automatisierung des Deployments. Die dafür notwendigen Schritte werden in diesem Kapitel beschrieben. Sie greifen nicht nur für RADAR Local, sondern betreffen ausdrücklich beide Angebotsformen.

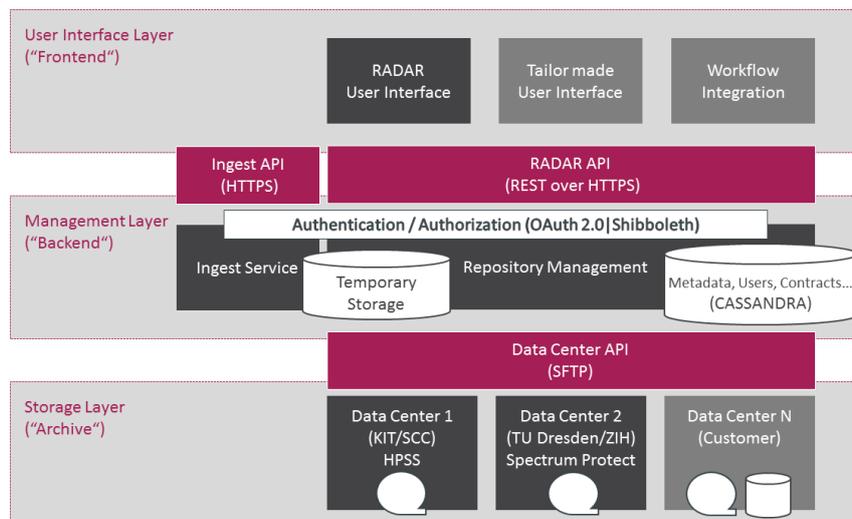


Abbildung 1: RADAR Systemarchitektur.

Das Aufsetzen und der Betrieb einer RADAR-Instanz sind komplexe Prozesse. Für die Bereitstellung von RADAR Cloud sind sowohl verschiedene interne Abteilungen bei FIZ Karlsruhe als auch bei externen Partnern (Steinbuch Centre for Computing<sup>3</sup> des KIT,<sup>4</sup> Zentrum für Informationsdienste und Hochleistungsrechnen der TU Dresden,<sup>5</sup> DataCite<sup>6</sup>) erforderlich. Die anfallenden Aufgaben umfassen grob die Bereiche Betriebssystem, Datenhaltung, Anwendung, Netzwerke und Monitoring. Jeder Bereich umfasst eine Vielzahl an einmaligen und wiederkehrenden Aufgaben (siehe Abbildung 2), die jeweils spezielles Domänenwissen benötigen. Jeder Bereich muss schnell auf ändernde Anforderungen bzw. Situationen reagieren können. Man denke zum Beispiel an das Einspielen von Sicherheitspatches und Bugfixes oder das Zurückspielen eines Backups. Diese Komplexität wird sich mit RADAR Local weiter erhöhen, denn mit der Installation jeder lokalen Instanz kommen weitere Parteien mit ins Spiel.

<sup>3</sup><https://www.scc.kit.edu/>

<sup>4</sup><https://www.scc.kit.edu/>

<sup>5</sup><https://tu-dresden.de/zih>

<sup>6</sup><https://datacite.org/>

Um Installationen und Betrieb für RADAR Cloud und mehrere Instanzen von RADAR Local parallel bewältigen zu können, benötigt FIZ Karlsruhe standardisierte und konfigurierbare Automatisierungswerkzeuge, die komplexe Aufgaben kapseln und so abteilungs- und institutionenübergreifende Arbeiten zusammenfassen und einfach bedienbar machen.

Betriebssystem	Datenhaltung	Anwendung	Netzwerk	Monitoring	Externe Abhängigkeiten
<ul style="list-style-type: none"> <li>• VMware</li> <li>• Paketverwaltung</li> <li>• System</li> <li>• Backup</li> <li>• NFS</li> </ul>	<ul style="list-style-type: none"> <li>• Datenbanken</li> <li>• Backup / Recovery</li> <li>• Suchindex</li> <li>• SFTP</li> </ul>	<ul style="list-style-type: none"> <li>• Applikationsserver</li> <li>• Konfiguration</li> <li>• Logging</li> <li>• Scheduler</li> </ul>	<ul style="list-style-type: none"> <li>• Firewall</li> <li>• Reverse Proxy</li> <li>• VPN</li> </ul>	<ul style="list-style-type: none"> <li>• Server</li> <li>• Datenbanken</li> <li>• Netzwerk</li> <li>• Anwendung</li> </ul>	<ul style="list-style-type: none"> <li>• DataCite</li> <li>• Archive (SCC, Dresden)</li> <li>• ORCID</li> <li>• CrossRef</li> </ul>

Abbildung 2: Notwendige Software-Komponenten für den Betrieb von RADAR Local.

## 3.2 Automatisierungswerkzeuge

Für RADAR Local liegt der Fokus der Automatisierung auf den Bereichen Systemkonfiguration, Deployment sowie Container-Virtualisierung und Container-Orchestrierung. Bei der Auswahl geeigneter Werkzeuge spielten Faktoren wie bereits vorhandene Kenntnisse und die Lernkurve eine wichtige Rolle. Die Entscheidungsfindung und untersuchte Alternativen zu den ausgewählten Werkzeugen sollen hier jedoch nicht weiter vertieft werden.

Ansible<sup>7</sup> ist ein Open-Source Automatisierungs-Werkzeug zur Orchestrierung, allgemeinen Konfiguration und Administration von Computern. FIZ Karlsruhe nutzt es in den Bereichen Systemkonfiguration und Deployment, spezifisch zum Aufsetzen und zur Wartung der eingesetzten Software-Infrastruktur. Das Tool zeichnet sich durch seine Einfachheit und die Unterstützung durch eine große Community aus. Ansible hat einen modularen Aufbau und unterstützt diverse Betriebssysteme. Eine wichtige Eigenschaft ist seine Idempotenz, d.h. auch bei mehrfacher Ausführung einer Aufgabe ist das gleiche Ergebnis wie bei einmaliger Ausführung erwartbar. Mit den Modulen für die CentOS-Paketverwaltung und Docker-Laufzeitumgebung können die Mitarbeiterinnen und Mitarbeiter von FIZ Karlsruhe einen Großteil der geplanten Aufgaben umsetzen. Ansible setzt auf der Zielinfrastruktur lediglich einen SSH Zugang voraus. Es integriert sich zudem in den Entwicklungsprozess und folgt der DevOps-Philosophie, die eine effektive und effiziente Zusammenarbeit zwischen Softwareentwicklung und IT-Betrieb anstrebt.

Automatisierungsaufgaben werden in sogenannten Ansible Playbooks gut lesbar formuliert und dienen so auch gleich der Dokumentation. Die dazugehörigen Host-Konfigurationen werden in einem Git-Repository für unterschiedliche Instanzen (Entwicklung, Test, RADAR Cloud, Universität Konstanz) abgelegt. Tabelle 1 zeigt die im Rahmen von RADAR Local entwickelten Playbooks mit deren Hilfe es nun möglich ist, innerhalb kurzer Zeit einen Cluster für den Betrieb von RADAR Local vorzubereiten. `setup-vm.yml` Installation von Basispaketen (`chronyd`, `net-tools`) Installation von Docker, Docker-Compose

<sup>7</sup><https://www.ansible.com/>

Tabelle 1: Ansible Playbooks für RADAR.

setup-vm.yml	Installation von Basispaketen (chronyd, net-tools) Installation von Docker, Docker-Compose Initialisierung http-proxy
init-docker-swarm.yml	Initialisierung des Docker Swarms
init-services.yml	Erstellen der Verzeichnisstruktur Kopieren von Konfigurationsdateien Starten der Dienste

Initialisierung http-proxy init-docker-swarm.yml Initialisierung des Docker Swarms init-services.yml Erstellen der Verzeichnisstruktur Kopieren von Konfigurationsdateien Starten der Dienste

Das Deployment der einzelnen Softwarekomponenten erfolgt in Containern. Container isolieren Anwendungen mit Hilfe von Container-Virtualisierung. Als Virtualisierungsumgebung kommt Docker<sup>8</sup> zum Einsatz, eine weit verbreitete freie Software, die auf vielen Linux-Distributionen eingesetzt werden kann und einfach zu erlernen ist. Für jede Anwendung oder Komponente von RADAR (siehe auch Abbildung 1) wird ein Docker-Image (die Basis eines Containers) verwendet. Ein Docker-Image kapselt den Dienst und alle vom Betriebssystem benötigten Ressourcen. Ein wichtiges Ziel beim Aufsetzen einer Docker-Imagefile ist es, die Konfiguration von der Software sauber zu trennen. Ebenso werden notwendige persistente Speicherbedarfe externalisiert. Dadurch lassen sich Images einfach ersetzen oder unverändert auf unterschiedlichen Systemen nutzen. Der Einsatz von Docker-Images spart zeitaufwendige Arbeitsschritte wie das Aufsetzen von Webservern, Datenbanken oder Suchindizes ein. Zudem stehen für viele der in RADAR eingesetzten Komponenten fertige und von der Community bereits umfänglich getestete Docker-Images bereit.

Docker Swarm<sup>9</sup> verknüpft die eingesetzten Container zu einer Gesamtanwendung und verteilt diese auf den Rechnern der Zielinfrastruktur. Docker Swarm hilft zudem bei der Skalierung und Lastverteilung (Load Balancing) von Applikationen.

### 3.3 Laufzeitumgebung

Für den Betrieb von RADAR Local stellt die Universität Konstanz via VMware mehrere virtuelle Maschinen zur Verfügung. Auf diesen wird seitens FIZ Karlsruhe mit Ansible die Docker-Laufzeitumgebung installiert. Anschließend wird der Docker Swarm initialisiert, indem eine Maschine zum Manager bestimmt und den restlichen Maschinen die Rolle

<sup>8</sup><https://www.docker.com/>

<sup>9</sup><https://docs.docker.com/engine/swarm/>

von Workern zugeteilt wird (siehe Abbildung 3). In einer Konfigurationsdatei (docker-compose file) sind die Beziehungen der einzelnen Dienste untereinander und damit die Struktur der Gesamtanwendung definiert. Der Swarm Manager liest diese Datei ein und verteilt daraufhin die einzelnen Dienste als Container auf den vorhandenen virtuellen Maschinen (Swarm Nodes).

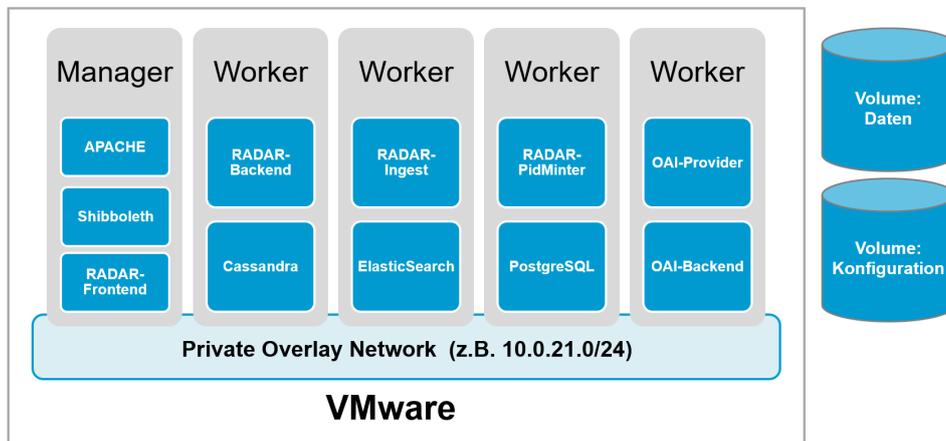


Abbildung 3: Docker Swarm-Umgebung.

Die Kommunikation zwischen den Services läuft in einem privaten Overlay-Netzwerk. Dieses virtuelle Netzwerk ist besonders geschützt und die gesamte interne Kommunikation erfolgt verschlüsselt. Dabei können gezielt einzelne Ports eines Containers, wie z.B. der Port 443 vom Webserver, für die VM freigegeben werden. Erst diese Portfreigabe ermöglicht Anfragen aus dem Internet via SSL.

Neben der Anwendung existieren noch zwei persistente Datenbereiche, die als Volumes in den Containern gemounted werden. Im Volume „Daten“ liegen die Datenbanken, der Suchindex, die Logdateien und die eingehenden Forschungsdaten. Im Volume „Konfiguration“ finden sich die Einstellungen für die einzelnen Dienste, wie z.B. die Webserverkonfigurationen oder Zertifikate.

### 3.4 Deployment Pipeline

Bei der Entwicklung und dem Betrieb von RADAR baut FIZ Karlsruhe auf die Prinzipien CI (Continuous Integration – mit jeder abgeschlossenen Änderung am Programmcode wird die Applikation neu kompiliert) und CD (Continuous Delivery – automatisierte Testung und Installation neuer Softwareversionen). Hierbei spielen die Entwickler regelmäßig ihre Quellcodeänderungen in das Software-Repositoryum Git ein. Auf einem Bamboo Build-Server werden die Komponenten automatisch gebaut und mit verschiedenen Tests und Qualitätssicherungswerkzeugen wie z.B. Sonar Cube oder DependencyCheck überprüft. Laufen diese Prozesse fehlerfrei durch, erfolgt anschließend die Installation der neu gebauten Software auf einer internen und einer externen Testumgebung.

Der Einsatz von Docker ermöglicht es, die Frequenz der produktiven Deployments von bisher ca. vier pro Jahr zu steigern und so näher an das Ziel einer echten Continuous Delivery zu kommen. Jede Softwareänderung (im Sinn eines neuen oder überarbeiteten Features oder eines Bugfixes) soll dabei automatisiert und schnell durch die Pipeline auf den Produktionsmaschinen installiert werden. Dies wird erst durch den Einsatz von Containertechnologie möglich, da nun in den Container-Images die benötigte Infrastruktur definiert und fertig getestet ausgeliefert werden kann und aufwändige manuelle Prozessschritte entfallen.

Die grün hervorgehobenen Zeilen in Abbildung 4 zeigen, in welchen Elementen der Deployment Pipeline FIZ Karlsruhe mit den vorbereitenden Schritten zur Einführung von RADAR Local Automatisierungswerkzeuge integrieren konnte. Die schwarz gesetzten Komponenten sind bereits seit längerer Zeit bei FIZ Karlsruhe im Einsatz.

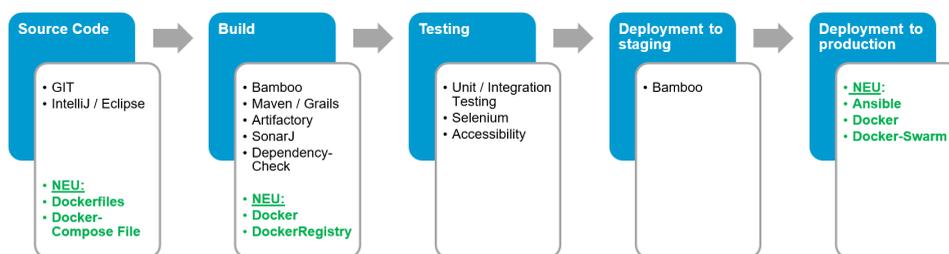


Abbildung 4: Continuous Integration / Continuous Delivery Pipeline.

### 3.5 Sicherheit

RADAR Local ist eine öffentlich zugängliche Webanwendung mit vielfältigen Schnittstellen zu anderen Systemen, zum Beispiel den eingebundenen Speichersystemen oder externen Diensten wie DataCite, CrossRef<sup>10</sup> und ORCID<sup>11</sup>. Neben der Härtung der Applikationskomponenten mittels automatisierter Tests mit OWASP ZAP (einem Open-Source Penetration Testwerkzeug) und der strikten Beschränkung der Kommunikation mit dem Internet auf HTTPS bzw. auf SFTP mit den Speichersystemen kapselt die Universität Konstanz ihre Instanz von RADAR Local in einem eigenen Subnetz, das vom Rest des Netzwerks der Universität und nach außen über Firewalls abgesichert ist.

Besonders sensibel ist der Zugriff der Administratoren von FIZ Karlsruhe über öffentliche Netze auf die Systeme der Universität Konstanz. Die gesamte Kommunikation zwischen FIZ Karlsruhe und den Systemen der Uni Konstanz erfolgt daher in einem verschlüsselten virtuellen privaten Netzwerk (VPN). Die Universität stellt darüber hinaus dedizierte Accounts für die Softwareinstallation und -pflege zur Verfügung.

<sup>10</sup><https://www.crossref.org/>

<sup>11</sup><https://orcid.org/>

Sämtliche Leistungen von FIZ Karlsruhe sowie die Mitwirkungspflichten einer RADAR Local nutzenden Einrichtung werden im „Dienstleistungsvertrag über die Nutzung von RADAR in kundeneigener Ablaufumgebung“<sup>12</sup> geregelt.

Informationen zu technischen Voraussetzungen und administrativen Vorarbeiten auf Institutionsseite sind als Checkliste<sup>13</sup> auf der RADAR-Website verfügbar.

## 4 Ausblick

Als Ergebnis der gemeinsamen Kooperation kann die Universität Konstanz mit dem fertigen Repository KonDATA zügig den gewünschten Service anbieten. Mit der reinen Bereitstellung von virtuellen Maschinen und Storage ist der Ressourcenaufwand auf der technischen Seite sehr gering. FIZ Karlsruhe kümmert sich um den Betrieb, die Wartung, die Pflege und die Weiterentwicklung. Als Dienstanbieter fungiert das KIM, das die IT-Infrastruktur bereitstellt, die Nutzenden unterstützt und die Daten kuratiert, Schulungen anbietet und Marketing für KonDATA betreibt.

Im Vergleich zu einer Eigenentwicklung wurde das gewünschte Ziel schneller und günstiger erreicht. Auch der dauerhafte Pflegeaufwand ist reduziert. Spannend bleibt die Frage, wie sich die Weiterentwicklung in Zukunft gestalten wird. Die Universität Konstanz ist Mitglied im Nutzerbeirat von RADAR geworden und kann somit aktiv Einfluss auf zukünftige Features und Verbesserungen nehmen. Ebenfalls herausfordernd wird die weitere Integration in die Dienstlandschaft an der Universität, da die Kontrolle über das System nicht vollständig in der eigenen Hand liegt. In der Gesamtschau überwiegen aber eindeutig die Vorteile. Die nicht vollständige Individualisierbarkeit wird durch ein nach dem Stand der Technik und mit geringem Aufwand betriebenes Repository mehrfach aufgewogen.

## ORCID IDs

- Matthias Landwehr  <https://orcid.org/0000-0001-9274-2578>
- Gabriel Schneider  <https://orcid.org/0000-0001-6573-3115>
- Stefan Hofmann  <https://orcid.org/0000-0003-0790-112X>
- Matthias Razum  <https://orcid.org/0000-0002-5139-5511>
- Kerstin Soltau  <https://orcid.org/0000-0002-6368-1929>

---

<sup>12</sup><https://www.radar-service.eu/index.php/de/nutzungshinweise>

<sup>13</sup>[https://www.radar-service.eu/sites/default/files/RADAR\\_Local\\_Checkliste.pdf](https://www.radar-service.eu/sites/default/files/RADAR_Local_Checkliste.pdf)