



netidee

PROJEKTE

RichPods.org

Endbericht | Call 19 | Projekt ID 7391

Lizenz CC BY-SA

Inhalt

1	Einleitung	3
2	Projektbeschreibung	3
3	Verlauf der Arbeitspakete	5
3.1	Arbeitspaket 1 – Projektstart	5
3.2	Arbeitspaket 2 – CI und UI-Design	5
3.3	Arbeitspaket 3 – RichPod-Server	6
3.4	Arbeitspaket 4 – RichPod-Editor	7
3.5	Arbeitspaket 5 – RichPod-Player	8
3.6	Arbeitspaket 6 – Podcast-Episoden	9
3.7	Arbeitspaket 7 – Dissemination	10
3.8	Arbeitspaket 8 – Projektabschluss	11
4	Umsetzung Förderauflagen	11
5	Liste Projektendergebnisse	12
6	Verwertung der Projektergebnisse in der Praxis	12
7	Öffentlichkeitsarbeit/ Vernetzung	13
8	Eigene Projektwebsite	13
9	Geplante Aktivitäten nach netidee-Projektende	14
10	Anregungen für Weiterentwicklungen durch Dritte	14

1 Einleitung

RichPods verfolgt das Ziel, ein innovatives Open-Source-Podcasting-Tool zu entwickeln, das Podcast-Episoden mit interaktiven Zusatzinhalten erweitert. Podcasting entstand als offenes Medium, wird jedoch zunehmend von großen Plattformen wie Spotify, Apple und Amazon in geschlossene Ökosysteme gedrängt, die Auffindbarkeit, Monetarisierung und Reichweite kontrollieren. RichPods setzt dem ein freies, erweiterbares System für interaktive und multimediale Podcasts entgegen – auf Basis offener Webtechnologien und ohne Plattformzwänge.

Im Projektverlauf wurden drei zentrale Softwarekomponenten umgesetzt: ein Cloud-native Server auf Basis von Node.js, ein umfangreicher Editor zur Erstellung von RichPods sowie ein Web-basierter Player zur Wiedergabe. Die Entwicklung erwies sich als deutlich komplexer als ursprünglich geplant, insbesondere beim Editor und Server, weshalb erhebliche Eigenleistungen über das Projektbudget hinaus erbracht wurden. Bei der Gestaltung wurde besonderer Wert auf Barrierefreiheit gelegt. Zusätzlich entstanden im Projektverlauf zwei weitere Komponenten für interaktive Karten und Diagramme, die als eigenständige Pakete auf NPM veröffentlicht wurden.

In Kooperation mit dem Nord.Post-Podcast wurde eine öffentliche Live-Aufnahme im Wiener Nordbahnhof durchgeführt, die wertvolle Praxiserfahrungen für das RichPod-Format lieferte.

Der gesamte Quellcode wurde unter der Mozilla Public License 2.0 auf GitHub veröffentlicht. RichPods wird nach Projektende als SaaS-Dienst unter richpods.org langfristig betrieben und weiterentwickelt, wobei in Monaten nach dem Launch eine Monetarisierungsstrategie für den nachhaltigen Betrieb angestrebt wird.

Der vorliegende Endbericht dokumentiert die Umsetzung im Rahmen vom Netidee Call 19 von Herbst 2024 bis März 2026. Wir beschreiben den Verlauf der Arbeitspakete, technische Entscheidungen, aufgetretene Herausforderungen sowie die geplanten Aktivitäten nach Projektende.

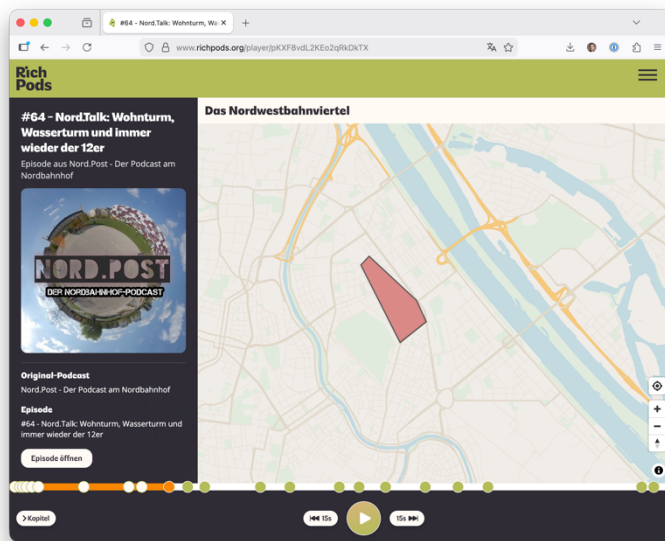
2 Projektbeschreibung

RichPods erweitert klassische Podcasts um zeitgesteuerte, interaktive Zusatzinhalte wie Karten, Infoboxen, Diagramme und Bilder, die synchron zur Audiowiedergabe eingeblendet werden. Wir setzen dabei konsequent auf offene Webtechnologien. Klares Ziel ist unabhängig von geschlossenen Plattformen wie Spotify oder Apple interaktive Podcasts anbieten zu können. Mit einem eigenen Editor, Player und Server entstand ein vollständiges Open-Source-System für eine multimediale Form des Podcastings.

Das Projekt versteht sich bewusst als Gegenentwurf zur zunehmenden Plattformisierung des Podcast-Ökosystems. Große Anbieter wie Spotify, Apple und Amazon drängen Podcasts in

geschlossene Systeme, kontrollieren Auffindbarkeit und Monetarisierung und schränken die Vielfalt ein. RichPods setzt stattdessen auf offene Webtechnologien und den RSS-Standard: Erstellte Inhalte bleiben frei zugänglich, und Podcaster*innen behalten die Kontrolle über Verbreitung und Verwertung.

Ein wesentlicher Teil vom Projekt war auch der eigentliche Betrieb von RichPods als SaaS-Angebot. Der Cloud-native Ansatz (im Konkreten auf Basis der Google Cloud Platform) ermöglicht einen besonders wartungsarmen und kostengünstigen Betrieb von RichPods als SaaS-Dienst. Infrastrukturaufgaben wie Skalierung, Verfügbarkeit und Sicherheitsupdates laufen weitgehend automatisiert ab. Durch den Einsatz von Cloud Run als Container-Plattform skaliert der Server dynamisch mit der tatsächlichen Nutzung, wodurch in der Anfangsphase kaum laufende Kosten anfallen und bei wachsender Nutzerbasis keine Architekturänderungen notwendig werden. Die Kombination aus Firestore als schemafreier NoSQL-Datenbank und Cloud Storage (S3-ähnlich) für unveränderliche Kapitel-Dateien erlaubt es, das Datenmodell flexibel weiterzuentwickeln, ohne aufwendige Datenmigrationen durchführen zu müssen. Gleichzeitig übernimmt die Google Identity Platform sicherheitskritische Aufgaben wie Passwort-Hashing und Token-Management, sodass kein eigener Authentifizierungsserver betrieben und gewartet werden muss. Um einen vollständigen Vendor Lock-in zu vermeiden, wurde das Datenmodell bewusst so gestaltet, dass eine Migration auf Firestore-kompatible Alternativen mit vertretbarem Aufwand möglich bleibt.



Für uns stellt der Projektabschluss eigentlich erst den richtigen Startschuss in das Projekt da, denn nun haben wir ein „MVP“ (Minimum Viable Product) fertig gestellt und gehen so richtig in die Öffentlichkeit. Wir möchten dafür sorgen, dass offene Standards und Web Technologien weiterhin die Grundlage für Podcasting bilden. Mit dem Projektende beginnt für uns die eigentliche Arbeit das Produkt „RichPods“ einer breiteren Masse zugänglich und bekannt zu machen.

Rückblickend auf den Projektablauf lässt sich sagen, dass es sehr schwierig ist in einem Antrag schon das komplette Produkt, wie wir es entwickelt haben, abzubilden. Wo am Anfang eine relativ einfache Idee niedergeschrieben wurde, ist es knapp zwei Jahre später ein durchaus interessantes Produkt entstanden. Die Entwicklung eines Produkts hat sich aber als deutlich umfangreicher dargestellt als die allerersten Ideen und Gedanken noch vermuten ließen. Dieser Mehraufwand und die zusätzlichen Schleifen einer Produktentwicklung haben sich gelohnt. Wir sind uns sicher, dass in den kommenden Wochen und Monaten zahlreiche RichPods angelegt werden und das Projekt auch eine Zukunft über den eigentlichen Projektzeitraum hat.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 – Projektstart

Zum Projektstart ergaben sich in diesem Arbeitspaket keine wesentlichen Auffälligkeiten. Die Projektwebseite einschließlich der zugehörigen Domain wurde eingerichtet, und die Projektdaten auf der Netidee-Seite wurden vollständig initial erfasst. Darüber hinaus wurden bis zum Projektabschluss sieben Blogbeiträge für die Netidee-Webseite erstellt. Ein weiterer Blogbeitrag ist noch in Planung.

Vor Beginn der eigentlichen Umsetzungsphase wurde das aktuelle Umfeld relevanter Podcast-Plattformen sowie der projektspezifische Kontext eingehend analysiert. Im Mittelpunkt standen insbesondere die Distribution von Podcasts, die Monetarisierung von Inhalten, Maßnahmen zur Eindämmung der fortschreitenden Zentralisierung der Podcast-Infrastruktur sowie Fragen der Bewerbung und des Marketings für Podcaster*innen.

Im Zuge dieser Umfeldanalyse wurde deutlich, dass eine Anpassung der Priorisierung der Arbeitspakete sinnvoll erscheint. Es wurde daher beschlossen, zunächst einen Player auf Basis von Prototyp- bzw. Dummy-Daten zu entwickeln und erst in weiterer Folge den Editor sowie den finalen Server umzusetzen. Aus diesem Grund wurde Arbeitspaket 5 vorgezogen und unmittelbar im Anschluss an Arbeitspaket 2 begonnen. Diese Änderung wurde Netidee am 12. Mai 2025 per E-Mail mitgeteilt.

3.2 Arbeitspaket 2 – CI und UI-Design

Für die Erstellung einer Basis-CI und eines UI-Entwurfs wurde Lisa Vietze extern beauftragt. Sie hatte bereits aus einem früheren Projekt Erfahrung mit der Netidee und kennt daher die grundsätzlichen Voraussetzungen.

Nach einigen Iterationen wurde eine erste Landing Page auf <https://www.richpods.org> bereitgestellt. Diese wurde noch um eine Team- und Kontakt-Seite erweitert. Bis zum öffentlichen Start des Players wurde sie noch um Kleinigkeiten und Projektinformationen erweitert. In den kommenden Wochen nach Projektabschluss wird die Webseite noch mit Meta-Inhalten zum Projekt ausgebaut werden.

Bei der Gestaltung haben wir besonders auf Barrierefreiheit nach WCAG 2.2 Level AA geachtet. Durch das Inkrafttreten des Barrierefreiheitsgesetzes (BaFG) und dem European Accessibility Act liegt nämlich besonderer Fokus auf diesem Thema. Dazu wurde auch ein eigener Blogbeitrag im Netidee-Blog verfasst: <https://www.netidee.at/richpodsorg/barrierefreiheit-gehoert-dazu>

3.3 Arbeitspaket 3 – RichPod-Server

Der ursprüngliche Zeitplan sah nur eine sehr kurze Entwicklungszeit vor. Dabei wurde von einer sehr einfachen Umsetzung ausgegangen, die früh im Projektverlauf abgeschlossen werden sollte. Dieser Plan hat sich so nicht in der Realität bestätigt. Die Entwicklung des Servers lief über den gesamten Projektzeitraum, da sich die Komplexität und Anforderungen laufend erhöht haben. Außerdem wurden deutlich mehr Funktionen in dieses Arbeitspaket eingebaut. Diese Mehraufwendungen wurden primär durch umfassende Eigenleistung über das Projektzeitbudget hinaus abgefangen.

Wir haben uns für einen Cloud-native Ansatz entschieden, da wir uns damit langfristige Vorteil im Betrieb und Wartung von RichPods als SaaS erhoffen. Der Server ist eine Node.js-Applikation (24+), die auf Express 5 basiert und eine GraphQL API zusammen mit mehreren REST-Endpunkten bereitstellt. Im Produktivbetrieb wird er als Container auf Google Cloud Run deployt. Verschiedene Service-Module stellen die Business Logic bereit, die wiederum per Resolver mit der GraphQL-Schnittstelle verbunden werden.

Bei der Auswahl der Datenbank war schlussendlich die einfache Wartung und ein langfristig sehr kostengünstiger Betrieb von RichPods als SaaS-Lösung der entscheidende Faktor.

Die Daten werden auf zwei Arten gespeichert:

- **Firestore** für klassische Entities. Der von Google Cloud bereitgestellte Firestore ist eine cloudbasierte NoSQL-Datenbank, die Daten in einer hierarchischen Struktur aus Sammlungen (Collections) und Dokumenten speichert. Dieses flexible Datenmodell ermöglicht es uns, ohne ein starres Tabellenschema laufend Erweiterungen durchzuführen und die laufenden Wartungsaufwände auf ein Minimum zu reduzieren.
- **Cloud Storage:** Einzelne Kapitel eines RichPods werden als unveränderliche JSON-Dateien in Cloud Storage gespeichert und im jeweiligen Kapitel-Dokument in Firestore referenziert.

Statt einer lokalen Postgres-Datenbank wurde eine Cloud-native Lösung auf Basis der Google Cloud Platform gewählt. Dadurch soll der laufende Betrieb kostengünstiger gestaltet werden und sich einfacher an unterschiedliche Nutzungsszenarien anpassen. Um keinen totalen Vendor Lock-in zu erhalten, wurde das Datenmodell so gestaltet, dass es relativ einfach auf Firestore-Alternativen migriert werden könnte. Es gibt keine direkte Open-Source-Implementierung von Firestore selbst, aber es existieren einige Alternativen, die ein ähnliches Datenmodell und ähnliche Funktionalität bieten.

Authentifizierung und Kontoverwaltung

Auch hier war eine möglichst wartungsarme und dauerhaft sichere Lösung für uns wichtiger als eine vollständig in unsere Hand liegenden Lösung. Für eine sichere Authentifizierung wurde die Google Identity Platform als Basis gewählt. Sie ist ein vollständig von Google verwalteter Authentifizierungsdienst, der unserem Projekt ermöglicht, sichere Nutzerverwaltung ohne

eigenen Auth-Server zu integrieren. Die Identity Platform unterstützt zahlreiche Anmeldemethoden, wobei wir zu Beginn nur E-Mail und Google ID unterstützen. Sicherheitskritische Aufgaben wie Passwort-Hashing und Token-Management werden automatisch übernommen, und die Plattform skaliert bei Projektwachstum ohne Anpassung der Architektur mit. Bis zu 10.000 aktiven monatlichen Nutzer*innen die die Nutzung kostenlos.

Sicherheitsaspekte

Hier mussten wir deutlich mehr Aufwand verzeichnen als ursprünglich im Antrag geplant. Da unsere Software prinzipiell für alle offen ist, haben sich auch neue potentielle Angriffsvektoren ergeben. Dieser Teil erklärt auch den gestiegenen Umsetzungsaufwand für den Server, da viel mehr Details und Szenarien betrachtet wurden.

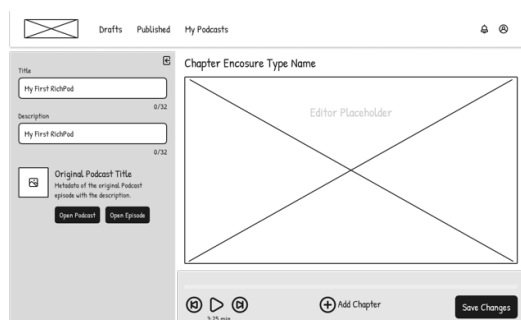
Alle Nutzereingaben werden vor der Verarbeitung mit Joi-Schemata validiert. Sicherheitsmechanismen auf API-Ebene verhindern HTML-Injection (noHtml), JSON-Prototype-Pollution (safeJson) und fehlerhaftes GeoJSON. Alle Schreibzugriffe unterliegen einem Rate Limiting. Alle Bild-Uploads werden validiert und zur Sicherheit neu generiert, MP3-Uploads per Batch-Skript validiert. Für beides gilt ein Größenlimit beim Upload. So möchten wir missbräuchliche Verwendung umfassend unterbinden.

3.4 Arbeitspaket 4 – RichPod-Editor

Dieses Arbeitspaket hat die größte Ausweitung vom ursprünglichen Projektplan erfahren. War der Editor anfänglich nur das Tool um Rohdaten in das RichPod-Datenschema zu gießen, wurde er im Endausbau ein vollumfängliches Podcasting-Tool, dessen Entwicklung kontinuierlich weitergeht und somit auch bis zum Projektende lief.

Über das Projektende hinaus sind noch einige UX-Nacharbeiten geplant, außerdem soll das Editieren von besonders langen Podcast-Folgen einfacher gestaltet werden.

Für den Editor wurde zuerst ein Wireframe-Entwurf erstellt. Dabei ist aufgefallen, dass viele UI-Komponenten zwischen Player und Editor geteilt werden müssen. Änderungen im Player hatten somit immer auch eine Auswirkung auf den Editor. Hier musste gerade an Anfang viel nachgebessert werden, was ein beträchtlicher Mehraufwand war. Schlussendlich haben wir aber eine deutlich einfachere Architektur ohne Doppellungen. Änderungen und neue Features lassen sich langfristig schneller in RichPods integrieren.



Wo wir beim Zwischenbericht einige Features nur als Rumpf bzw. nur als MVP-Variante gesehen haben, ist uns dennoch eine vollständige oder sogar erweiterte Umsetzung gelungen. Am Ende hat der Editor einen deutlich größeren Funktionsumfang als wir noch zum Antragszeitpunkt dachten. Es können nicht nur einzelne Kapitel bearbeitet werden, sondern auch ganze Podcast-Episoden selbst gehostet werden. Außerdem laden wir automatisch Vorschauen für Links herunter, extrahieren Daten aus Wikidata und haben das komplette UI für Lokalisierungen vorbereitet. Standardmäßig ist der Editor in Deutsch und Englisch übersetzt.

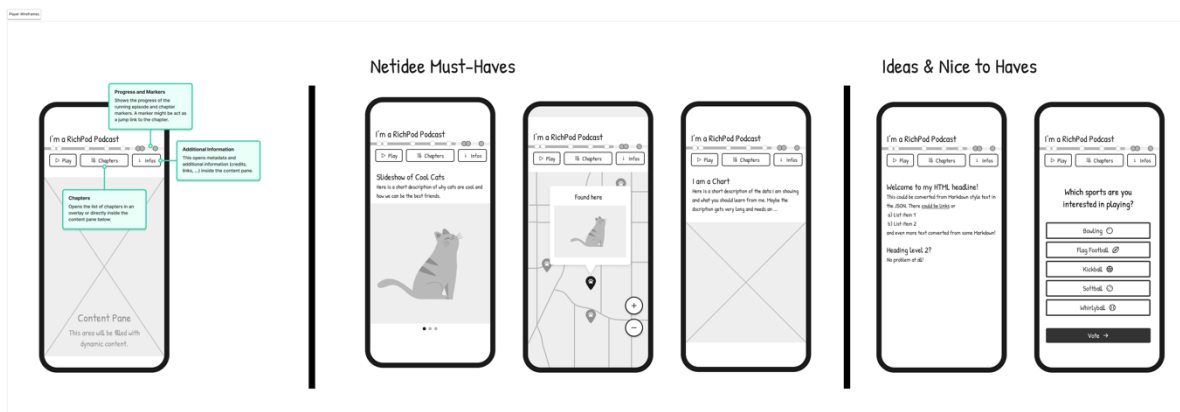
Statt externe Komponenten einzusetzen, haben wir für interaktive Karten und für Diagramme eine eigene separate Vue-Komponente entwickelt. Diese sind als eigenständige Pakete von uns auf NPM veröffentlicht worden und können auch in anderen Projekten eingebunden werden. Beide Komponenten können direkt im Web ausprobiert werden:

- <https://code.richpods.org/tiny-geojson-tool/>
- <https://code.richpods.org/tiny-spreadsheet-editor/>

3.5 Arbeitspaket 5 – RichPod-Player

Ursprünglich sollte der Player nachgeordnet entwickelt werden und Ende 2025 für alle online verfügbar sein. Dieser Plan hat sich bis in den Februar 2026 geschoben, wo wir einen „Soft Launch“ vornahmen und Beispiel-RichPods online stellten. Der Entwicklungszeitraum vom Player ist ebenfalls deutlich länger ausgefallen, da sich der Funktionsumfang deutlich erweiterte.

In einem ersten Schritt wurden Basis-Wireframes erstellt. Grundlage bildete eine Umschau unter diversen Podcast-Playern am Markt, u.a. Spotify, Apple Podcasts, Overcast, Audible, Google Podcasts, Castbox und weitere Web-basierte Player.



Diese Wireframes bildeten die Grundlage für die weitere Umsetzung.

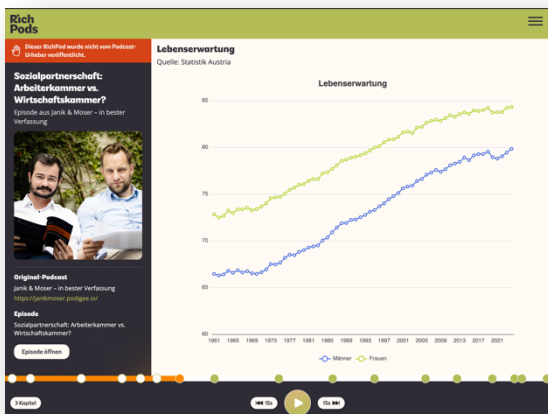
Eine unerwartete Herausforderung war die Synchronisation einzelner UI-Komponenten mit dem jeweiligen Abspielzustand der Podcast-Audiodatei. Die Architektur der Frontend-Applikation musste hierfür nachträglich noch einmal adaptiert werden, damit alle Audio-Events und -Zustände auch korrekt in den UI-Komponenten aufscheinen.

Ein weiterer Mehraufwand entstand durch die Anforderung, dass ein RichPod auch auf einer beliebigen Webseite integriert werden soll können. Dafür war ebenfalls eine Adaptierung der Grundarchitektur vom Frontend notwendig, die aber recht zügig umgesetzt werden konnte.

Basierend auf dem Grundlayout aus AP 2 wurde im August 2025 noch eine leichte Adaptierung am UI vorgenommen. Wir haben gemerkt, dass unsere Komponenten zwar gut auf Mobilgeräten funktionieren, aber wesentliche Schwächen im Desktop-View hatten. Eine stärkere Beachtung von Desktop-Ansichten schon im Designprozess hätte hier den leichten Mehraufwand verhindert. Wir haben nachträglich in Absprache mit unserer Designerin das Desktop-Layout erstellt und in den Player schrittweise integriert. Somit funktionieren RichPods auf allen gängigen Geräten, auch wenn wir noch zahlreiche Optimierungsmöglichkeiten entdeckt haben. Diese sollen nach Projektabschluss fortgesetzt werden, nachdem unser SaaS-Service startet.

3.6 Arbeitspaket 6 – Podcast-Episoden

Gerade in diesem Bereich war es zum Zeitpunkt der Antragstellung schwer, einzelne Maßnahmen und zusätzliche Aktivitäten vorab festzulegen. Der Zeitplan für dieses Arbeitspaket war bewusst sehr breit gefasst, da hier journalistische Inhalte entstehen und diese dann in unserem Editor aufbereitet werden sollten. Wir haben aber im Laufe des Herbst 2025 entschieden, dass wir erst mit einer wirklich stabilen Version von RichPods die Umsetzung durchführen können. Sobald sich nämlich das Datenmodell im Server ändert, muss die komplette Folge neu produziert werden und das wäre ein hoher Mehrfachaufwand. Somit wurde laufend das Material gesammelt und ab Februar 2026 in RichPod-Form umgesetzt.



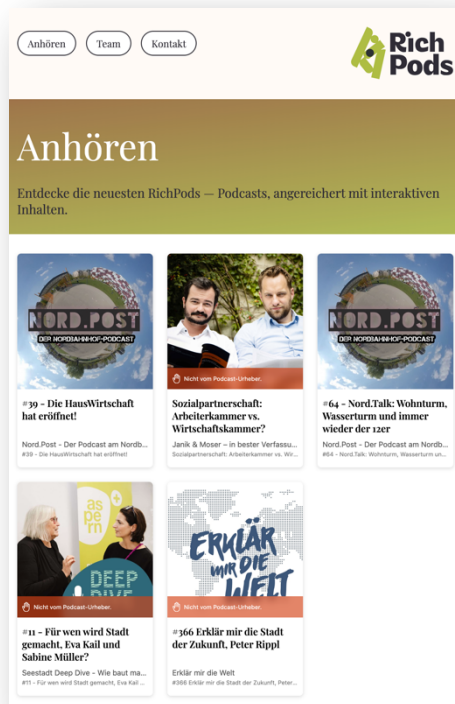
Sonja Harter hat jedoch über das gesamte Jahr 2025 bei allen Vorproduktionen von Episoden entsprechendes Material für RichPod-Elemente gesammelt. Dieser Prozess lieferte zudem wertvolle Inputs für den weiteren Entwicklungsprozess des Players, da erst dadurch klarer wurde, welche Anforderungen in der Praxis tatsächlich auftreten.

In den ersten Projektmonaten war regelmäßiges Brainstorming wesentlich für den Projektfortschritt. In der Entwicklung des

Players wurden wiederholt Feedback-Loops eingebaut. In agilen Methoden ausgedrückt trat Sonja Harter dabei als Product Owner auf, für die das Produkt schlussendlich gebaut wird.

Im Jänner 2026 realisierte RichPods in Kooperation mit dem Lokalmedium Zwischenbrücken und dem Nord.Post-Podcast eine Live-Aufnahme im NordbahnSaal der HausWirtschaft in Wien – als Ersatz für die ursprünglich geplante Kandidat:innen-Runde zur vorgezogenen Wien-Wahl. Thema

der Podiumsdiskussion war der Leerstand im Nordbahnviertel. Mit rund 60 Besucher:innen stieß die Veranstaltung auf gute Resonanz aus dem Viertel und den umliegenden Stadtteilen. Technisch konnten beim Live-Recording sechs Tonspuren und zwei Videospuren gleichzeitig von einem einzigen Mischpult aus aufgezeichnet werden, was auch Publikumsfragen gut dokumentierbar machte. Die gewonnenen Aufnahmen bilden die Grundlage für eine RichPods-Episode und lieferten wertvolle Praxiserfahrungen für den Einsatz des RichPod-Formats im Live-Kontext.



In den Tagen vor der finalen Endberichtslegung im März 2026 ist ein Bug in der Darstellung der Link-Cards und den versteckten Kapiteln aufgefallen. Um nicht mehrfach einen RichPod neu anlegen zu müssen oder sogar mit einem Migrationsskript die Daten anpassen zu müssen, haben wir uns entschlossen, drei NordPost-Folgen in RichPod-Form erst nach dem formalen Projektende zügig nachzuliefern. Dieser Umstand wurde auch dem Netidee-Team mitgeteilt und der Fertigstellungsgrad von AP 6 ist zum Endbericht nur 95%. Bis April 2026 werden aber alle drei fehlenden NordPost-Folgen auch als RichPod online gehen. Dennoch sind zum Endbericht in Summe fünf RichPods online gegangen – und es werden laufend mehr!

3.7 Arbeitspaket 7 – Dissemination

Wir konnten mit diesem Arbeitspaket nur sehr spät rund um den Jahreswechsel 2025/26 beginnen, da davor die öffentliche Variante noch nicht verfügbar war. Deshalb hat sich das Informationsmaterial auf ein Minimum beschränkt und wir haben Beispiele nur in der Entwicklungs-Instanz demonstriert. Ein kurzes Whitepaper wurde verfasst, außerdem eine Produktpräsentation.

RichPods wurde in einem fortgeschrittenen Stadium folgenden Personen gezeigt zwei bekannten Podcast-Hosts aus Österreich gezeigt. Ihr Feedback wird nach Projektende in die weitere Entwicklung einfließen.

Da der SaaS-Betrieb mit 20. März startete, werden einige zu diesem Arbeitspaket thematisch passende Tätigkeiten erst nach dem Projektende umgesetzt werden. Es wurden bereits kleinere Werbematerialien (Sticker, Visitenkarten) für unsere SaaS-Lösung vorbereitet, die mit dem Launch bestellt werden.

3.8 Arbeitspaket 8 – Projektabschluss

Der geplante Projektabschluss mit Ende Februar 2026 konnte nicht gehalten werden. Wir haben am 5. Februar 2026 eine Verlängerung auf den 20. März 2026 vom Netidee-Vorstand angefragt, die genehmigt wurde. Somit blieb uns genug Zeit, auch alle Unterlagen und Dokumentationen noch ordentlich und umfassend fertigzustellen.

Am 10. März 2026 wurde das öffentliche Github-Repository mit dem ersten öffentlichen Commit gefüllt. Darin enthalten waren alle Komponenten von RichPods. Am 20. März wurde die Webseite erweitert und mit den ersten RichPods befüllt. Wir planen für Ende März einen öffentlichen Launch, bei dem unser SaaS-Service bekannter gemacht wird.

Im Rahmen von Arbeitspaket wurden alle formalen Abschlussaktivitäten des Projekts durchgeführt. Dies umfasste die Erstellung der Endabrechnung, der Entwickler*innen- und Anwender*innendokumentation sowie des Endberichts und der Zusammenfassung. Darüber hinaus wurde ein abschließender Blogbeitrag verfasst und alle relevanten Dokumente an die Netidee übermittelt.

4 Umsetzung Förderauflagen

Nicht auf RichPods zutreffend, es gibt keine besonderen Förderauflagen.

5 Liste Projektergebnisse

1	Projektwischenbericht	CC BY-SA 4.0	https://www.netidee.at/richpodsorg
2	Projektendbericht	CC BY-SA 4.0	https://www.netidee.at/richpodsorg
3	Entwickler_innen-Dokumentation	CC BY-SA 4.0	https://www.netidee.at/richpodsorg
4	Anwender_innen-Dokumentation	CC BY-SA 4.0	https://www.netidee.at/richpodsorg
5	Einseitiges Whitepaper und Kurzübersicht	CC BY-SA 4.0	https://www.netidee.at/richpodsorg
6	Dokumentation Kommunikation	CC BY-SA 4.0	Im Endbericht; https://www.netidee.at/richpodsorg
7	RichPod Server (Software-Server)	Mozilla Public License 2.0	Im Mono-Repository unter https://github.com/richpods/richpods/
8	RichPod Editor (Software-Client)	Mozilla Public License 2.0	Im Mono-Repository unter https://github.com/richpods/richpods/
9	RichPod Player (Software-Client)	BSD	Im Mono-Repository unter https://github.com/richpods/richpods/
10	Podcast-Episoden	CC BY-SA 4.0	Auf RichPods.org und im Blog auf https://www.netidee.at/richpodsorg

6 Verwertung der Projektergebnisse in der Praxis

Wir starten mit dem Projektende in den öffentlichen SaaS-Betrieb von RichPods.org und damit den öffentlichen Auftritt. Damit ist eine Weiterentwicklung der Software sichergestellt.

Der NordPost-Podcast wird auch in Zukunft mit RichPods-Episoden arbeiten. Natürlich ist unser Ziel, dass so viele wie möglich Podcast-Creators unser Tool zumindest als SaaS nutzen.

Durch die Cloud-native Implementierung der Software sollte es auch für andere möglich sein, eigene RichPods-Instanzen zu erstellen. Wir sind hier noch am Überlegen, wie diese besser dabei unterstützt werden können.

7 Öffentlichkeitsarbeit / Vernetzung

Da wir erst im März 2026 mit der SaaS-Variante gestartet sind, gab es nur eine rudimentäre Vernetzungsarbeit bisher. Im ursprünglichen Projektplan wäre die Veröffentlichung deutlich vor dem Projekt Ende erfolgt, das ist sich aber nicht ausgegangen.

Wir haben vor allem ab Sommer 2025 mit mehreren Podcast-Hosts gesprochen und ihnen unsere Idee vorgestellt. So konnte auch schon etwas Feedback in unser Produkt einfließen.

Die öffentliche Aufzeichnung im NordbahnSaal vor rund 60 Menschen wurde auch offiziell von RichPods unterstützt und wir waren in Form vor Rollup und entsprechenden Erwähnungen präsent. Die Nachbericht dazu haben wir im Blog veröffentlicht:

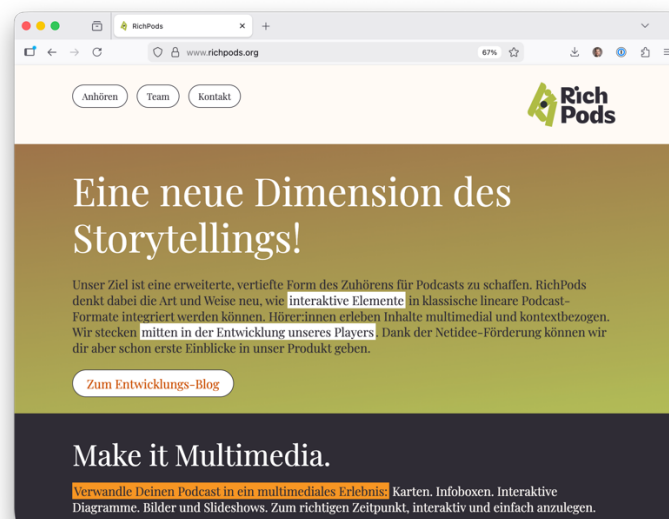
<https://www.netidee.at/richpodsorg/air-richpods-goes-nordbahnsaal>

8 Eigene Projektwebsite

Wir betreiben RichPods als SaaS-Modell unter <https://www.richpods.org>

- Der Editor selbst kann unter <https://www.richpods.org/editor/> aufgerufen werden.
- Einen Überblick über publizierte RichPods gibt es unter <https://www.richpods.org/anhoeren/>

Leider war es uns nicht möglich, die Domain richpods.com zu erhalten. Diese würde nur für eine relativ hohe Ablösesumme von rund 5.000 Euro an uns übergehen können.



9 Geplante Aktivitäten nach netidee-Projektende

Das Projekt wird unter <https://www.richpods.org/> als SaaS bereitgestellt und weiter ausgebaut. Eigentlich wollten wir bereits in der Projektphase intensiver mit der Community sprechen, allerdings wurden unser Editor und Player erst im Jänner 2026 soweit fertig, dass wir ihn tatsächlichen User:innen zeigen könnten. Daher geht es in den kommenden Monaten vor allem um die Bewerbung unseres Dienstes und es wird weitere Entwicklungsschritte geben.

Sollte es einen Netidee Call 21 geben, werden wir jedenfalls um eine Anschlussförderung ansuchen, um unser Projekt noch deutlich erweitern zu können.

Langfristig muss eine Monetarisierungsstrategie gefunden werden, um den laufenden Betrieb und die Weiterentwicklung zu finanzieren. Wir denken auch an Custom-Hosting von RichPods für größere Medienanbieter und den Ausbau des Hosted Services.

10 Anregungen für Weiterentwicklungen durch Dritte

Wir hoffen, dass gerade der Player von Dritten aufgegriffen wird und die Integration in z.B. CMS-Systeme erfolgt. Damit könnte die Verbreitung von RichPods schneller erfolgen.

Content-seitig möchten wir durch Demos und direkte Gespräche für eine Adaption unserer Idee in populären Podcasts sorgen.

