



Digital Transformation Success by AppNavi

PRODUKTBESCHREIBUNG

Stand: 1. Dezember 2023



INHALT

1	Über diese Produktbeschreibung	4
2	Was ist AppNavi?	4
2.1	Vorteile durch AppNavi.....	5
2.2	Unterstützte Systeme & Oberflächentechnologien.....	5
2.3	Unterstützte Kundensituationen	5
2.4	Integrationsstrategien	6
2.5	Kompatibilität	6
2.6	Leistungsbereiche.....	6
2.7	Produktkomponenten	7
2.8	Nutzertypen	7
3	Architektur / Komponenten.....	8
3.1	Tenant.....	8
3.1.1	Enterprise Authentication	8
3.1.2	Experten-Modus.....	8
3.1.3	Medienbibliothek.....	8
3.1.4	Userverwaltung.....	9
3.2	Workspaces	9
3.3	Apps.....	9
3.3.1	Einstellungen.....	9
3.3.2	Customizing Optionen	10
3.4	Contents.....	10
3.4.1	Versionierung.....	10
3.4.2	Zuordnung zu einer App.....	10
3.4.3	Kopieren von Contents	10
3.4.4	Veröffentlichung in einer App.....	10
3.5	Rollen & Berechtigungen.....	11
4	Routen.....	12
4.1	Darstellung.....	12
4.2	Aufbau & Bestandteile.....	12
4.3	Erstellen von Routen	13
4.4	Elementerkennung	14
4.5	Trigger & Modi	14
4.6	Varianz & Lernen	14
4.7	Automatisierung & RPA	15
4.8	Versionierung & Publikation	15
4.9	Sichtbarkeit und Segmentierung.....	16
4.10	Weitere Features.....	16
5	Neuigkeiten & Ankündigungen	18
5.1	Darstellung.....	18
5.2	Aufbau & Bestandteile.....	18
5.3	Trigger & Modi.....	19
6	Hotspots.....	20
6.1	Darstellung.....	20



6.2	Trigger & Modi	20
7	Trainings & Kollektionen	21
8	Analytics.....	22
8.1	Datenerfassung	22
8.2	Metriken.....	22
8.3	Diagramme	23
8.3.1	Allgemein.....	23
8.3.2	KPIs zum Workspace.....	23
8.3.3	Benutzeraktivität	23
8.3.4	Meistgenutzten Anwendungen.....	24
8.3.5	Inhaltsverwendung.....	25
8.3.6	Aufgewendete Zeit	25
8.3.7	Übersicht der Anwendungen	26
8.3.8	Verwendete Browser	26
8.3.9	Verwendete Sprachen	27
8.3.10	Routenprobleme.....	27
8.3.11	KPIs zu einer Route	27
8.3.12	Routenfortschritt	28
8.3.13	KPIs zu einer News bzw. einer Ankündigung	28
8.3.14	Neuigkeiten	28
9	User Behavior Mining.....	30
9.1	Verwalten von User Behavior Mining-Kampagnen.....	30
9.2	Interaktions-Objekte	30
9.3	Sessions.....	31
9.4	Behavior Flows.....	31
9.5	Automatisierte Erstellung von Routen	31
9.6	Datensicherheit.....	31
10	Möglichkeiten für Customizing.....	32
10.1	Standard Text Styling.....	32
10.2	Custom Theming.....	32
10.3	Custom Coding.....	33
11	Datenschutz	35
12	IT-Security.....	35
13	Software-Ergonomie.....	35
14	Grenzen und Limitierungen.....	36
15	Weitere Leistungsbereiche.....	37



1 Über diese Produktbeschreibung

Wir freuen uns über Ihr Interesse an AppNavi. Mit der vorliegenden Produktbeschreibung möchten wir Ihnen einen Überblick über die Funktionalitäten von AppNavi geben.

Da wir AppNavi kontinuierlich an den Bedürfnissen unserer Kunden weiterentwickeln, kann es zu temporären Abweichungen zwischen der Beschreibung und dem Produkt kommen – wir behalten uns Änderungen in der Beschreibung sowie dem Produkt entsprechend vor.

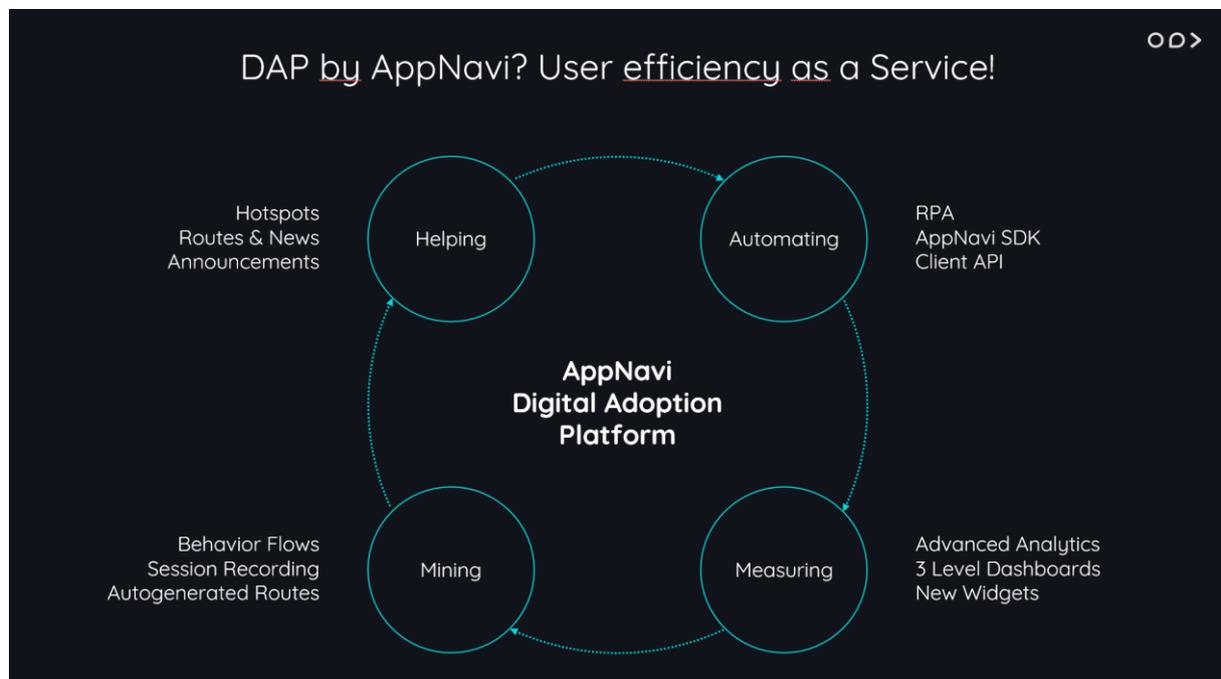
2 Was ist AppNavi?

AppNavi ist eine Digital Adoption Plattform. Der besondere Fokus von AppNavi ist die Unterstützung von großen Corporate Lösungen. Als die „am einfachsten zu nutzende Digital Adoption Plattform“ lässt sich AppNavi sehr einfach in der Fläche durch einzelne IT-Projekte oder große Transformationsprogramme nutzen.

Unter „einfach“ verstehen wir jeden Aspekt in der Arbeit mit unserer DAP. Hier stehen neben den Endanwendern auch die Autoren sowie alle weiteren Rollen innerhalb einer professionellen Unternehmensorganisation (wie bspw. IT und Betriebsrat) im Fokus.

Mit AppNavi können entweder einzelne Applikationen oder ganze Software-Landschaften unterstützt werden. Die flexible Architektur von AppNavi erlaubt es hierbei jede gewünschte Kundenorganisation abzubilden und Endanwender bestmöglich zu unterstützen.

Die Plattform setzt sich aus 4 Leistungsbereichen sowie unterstützenden Features zusammen:





2.1 Vorteile durch AppNavi

Gerade in komplexen oder nur selten genutzten Anwendungen fühlen sich Nutzer oft verloren. Im Ergebnis vergeht viel Zeit durch unnötige Trainings, vergebliche Versuche oder teure Anfragen an Kollegen oder das Servicedesk.

An dieser Stelle unterstützt AppNavi, indem es Anwender befähigt, unterstützt und beschleunigt. Mitarbeiter werden „on the Job“ trainiert. Dadurch reduzieren Sie Trainingskosten, da das Training zum Zeitpunkt des Problems erfolgt und nicht auf ein flüchtiges Vorratswissen abzielt. Notwendiges Toolwissen & Prozesswissen wird „just in time“ verfügbar gemacht. AppNavi wird im Rahmen von Digitalisierungsinitiativen eingesetzt und erzielt quantitative und qualitative Ergebnisse.

Quantitative Ergebnisse

- > Reduzierte Trainingskosten
- > Gesteigerte Effizienz der Endanwender
- > Niedrigere Support Aufwände
- > Höhere First-Time-Right
- > Höhere Conversion

Qualitative Ergebnisse

- > Unterstützung von strategischen Maßnahmen wie bspw. Kostensenkungsprogramme, M&A Prozesse, Expansionsstrategien, etc.
- > Verbesserte User Centricity
- > Höhere Akzeptanz

2.2 Unterstützte Systeme & Oberflächentechnologien

AppNavi kann in jede browserbasierte Software integriert werden und unterstützt sowohl HTML wie auch SVG Elemente.

Je nach Integrationsstrategie kommen hier sowohl eigens entwickelte Applikationen wie auch SaaS Lösungen in Frage. Häufige Anwendungsfälle sind:

- > Employee Self-Service Portale
- > Collaboration Systeme
- > BPM, ERP & CRM Lösungen
- > Reporting Apps & Dashboards
- > Customer Apps

Hier einige Beispiele: SAP SuccessFactors, SAP Ariba, Salesforce, Office 365 | M365, SharePoint, SharePoint Online, Outlook Online, Monday.com, Jira, Confluence, ServiceNow, Trello, Bitrix24, COYO, Slack, Workday, MicroStrategy, usw.

2.3 Unterstützte Kundensituationen

AppNavi unterscheidet drei typische Kundensituationen:

- > Endkunden
- > SaaS-Anbieter
- > Beratungen

AppNavi unterstützt Anwender in webbasierter Software. Je komplexer diese ist, umso besser können die Stärken von AppNavi ausgespielt werden. Der klassische Anwendungsfall für AppNavi ist daher ein Unternehmen, welches seine intern oder auch extern genutzte Anwendungslandschaft unterstützen möchte. Mitarbeiter, Bewerber, Kunden sowie Interessenten dieses Kundenunternehmens werden mit Hilfe von AppNavi durch die Systeme geführt.

Neben den Unternehmen selbst, können auch Anbieter von SaaS-Lösungen AppNavi einsetzen. Hier wird die Unterstützung zentral durch den Anbieter erstellt und an alle Kunden des Anbieters ausgeliefert.



Neben SaaS-Anbietern können auch Trainingsanbieter oder Beratungen Inhalte für ausgewählte Applikationen anlegen und an Ihre Kunden ausspielen.

In den beiden letztgenannten Kundensituationen können neben den zentral zur Verfügung gestellten Inhalten auch pro Kunde jeweils eigene Inhalte erstellt und angelegt werden.

2.4 Integrationsstrategien

AppNavi wird clientseitig in eine bestehende browserbasierte Software integriert. Dies erfolgt zentral seitens der Kunden-IT. Endanwender müssen nichts unternehmen – für sie wird AppNavi ohne ihr Zutun sichtbar. Hinsichtlich der Integration von AppNavi lassen sich zwei Strategien unterscheiden:

- > Direkte Integration
- > Integration per Browser-Erweiterung

Beide Strategien führen zu einer Client-seitigen Integration von AppNavi in die Zielapplikation. Bei der direkten Integration wird eine JavaScript Referenz in die Zielapplikation eingebettet. Falls Sie AppNavi innerhalb einer Organisation gleichzeitig in verschiedenen Applikationen nutzen möchten oder falls Sie keinen Zugriff auf die Zielapplikation haben, steht eine Integration per Browser-Erweiterung zur Verfügung. Diese versetzt den Browser in die Lage, AppNavi innerhalb einer Applikation nutzen zu können. Unsere Browser Erweiterungen werden per Group Policy ausgerollt und sind ebenfalls für Endanwender sofort und ohne Zutun nutzbar.

2.5 Kompatibilität

AppNavi funktioniert als Navigations-System für alle browserbasierten Anwendungen. Dabei ist völlig unerheblich, auf welchem Gerätetyp (Laptop, Tablet oder Handy) bzw. mit welchem Betriebssystem (Windows, Mac OS, Linux, etc.) Ihre Nutzer AppNavi nutzen möchten. Für mobile Geräte sind i.d.R. Anpassungen vorzunehmen, welche im initialen Integrationsprojekt berücksichtigt werden.

Die Objekterkennung von AppNavi funktioniert auf HTML und SVG Elementen sowie in allen zugrundeliegenden Frameworks (Vanilla, React, Vue, Angular, etc.).

Eine (nicht abschließende) Liste der unterstützten Browser finden Sie hier: <https://docs.appnavi.eu/docs/browser-compatibility>

2.6 Leistungsbereiche

AppNavi besteht aus insgesamt 4 Leistungsbereichen. Aufgrund der Architektur von AppNavi greifen diese auf intelligente Art und Weise ineinander und unterstützen sich somit gegenseitig.

Damit Endanwender ein System bestmöglich nutzen können, stehen mit AppNavi Features für Hilfe & Information live im Arbeitsfluss zur Verfügung. So können Oberflächen und Prozesse erklärt werden oder auch wichtige Informationen live und situativ gesteuert in den Arbeitsablauf der User gebracht werden. Diese Features kommen häufig im Rahmen von OnBoarding Szenarien zum Einsatz.

Um Anwender auch nach einem OnBoarding bestmöglich unterstützen zu können, stehen zahlreiche Features aus dem Bereich RPA (Robotic Process Automation) zur Verfügung. Hier liegt der Fokus der Anwendung nicht auf dem Training, sondern auf der beschleunigung von Anwendern in Ihrer täglichen Arbeit.

Um die Wirksamkeit der umgesetzten Maßnahmen bewerten zu können, steht ein ausgereiftes Set an Analyserwerkzeugen zur Verfügung. Hier kann die Nutzung der durch AppNavi bereitgestellten Inhalte kontrolliert – und Maßnahmen daraus abgeleitet werden.

Neben der einfachen Bereitstellung und Auswertung möglicher Maßnahmen im Bereich Digital Adoption, soll AppNavi seine Kunden auch bei der Planung unterstützen. Das User Behavior Mining liefert hier exakte Einblicke in die Nutzung der Ziel-Software durch die Anwender.

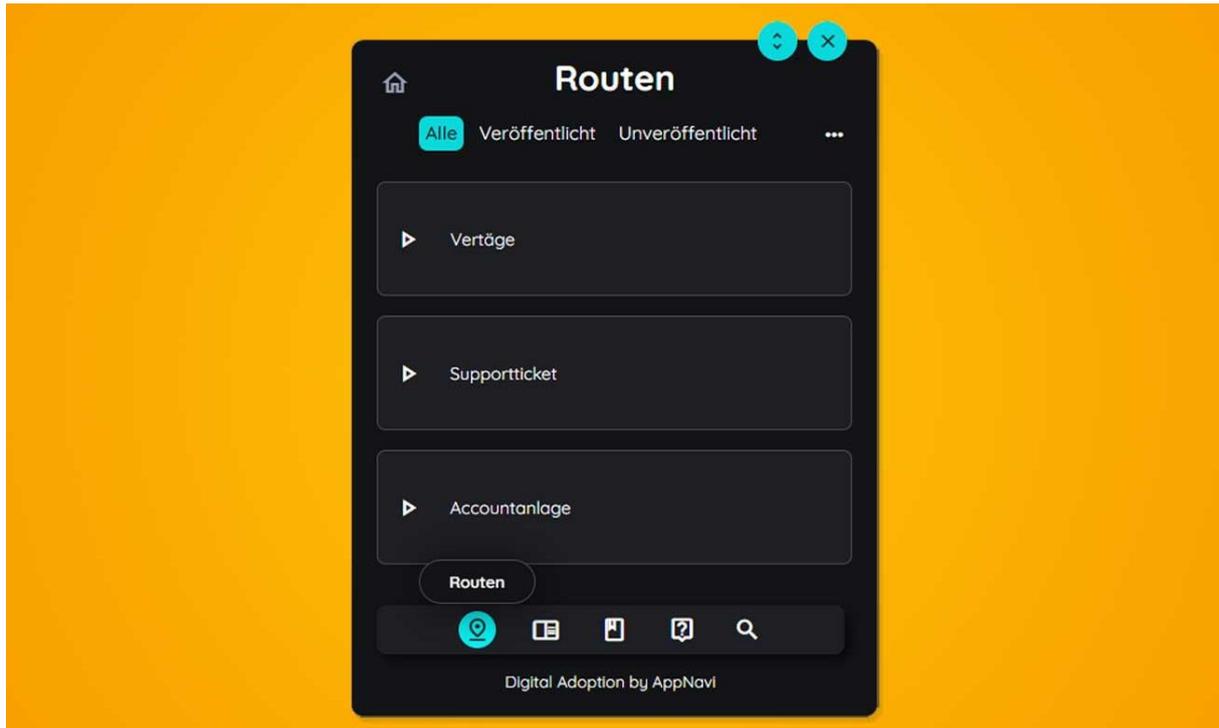
Ergänzend zu den 4 Kernleistungsbereichen stellt AppNavi auch eine Reihe weiterer Features zur Verfügung. Beispielsweise ermöglicht ein Content Store den einfachen Bezug bereits fertiger Inhalte über die AppNavi Community.

2.7 Produktkomponenten

AppNavi bildet jeden Kunden in einem logisch getrenntem Tenant ab. In diesem können alle Ihre Apps, User, Inhalte etc. verwaltet werden. Der Tenant kann über das **Kundenportal** aufgerufen werden. Hierbei handelt es sich um eine Website, welche über den Browser aufgerufen werden kann.

Mit Hilfe einer angelegten App wird das **AppNavi Widget** innerhalb des Zielsystems sichtbar. Unser Widget ist die visuelle Darbietung von AppNavi für Ihre Nutzer innerhalb Ihrer Systeme.

Je nach Rolle kann ein User innerhalb des Widgets den **Planer & Rekorder** von AppNavi öffnen. Dieser steht für die Bearbeitung und Erstellung von Routen zur Verfügung. Mit Hilfe einer grafischen Benutzeroberfläche können Autoren sehr einfach und ohne IT-Knowhow Routen anlegen.



2.8 Nutzertypen

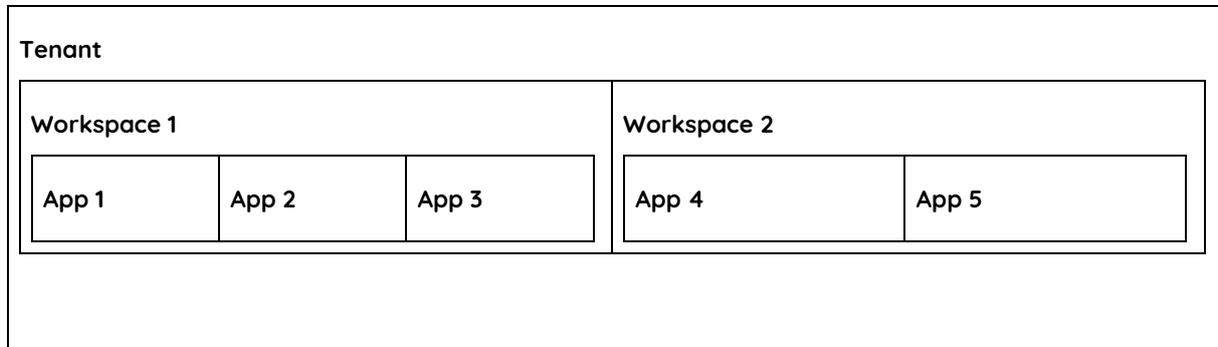
AppNavi unterscheidet zwischen Endanwendern und Autoren. Endanwender werden durch AppNavi unterstützt und können hierfür die Inhalte nutzen, welche durch Autoren zur Verfügung gestellt werden. Bei der Nutzung von AppNavi gibt es also eine große Gruppe der Endanwender sowie eine kleine Zahl an Autoren.

Autoren haben einen Login zu AppNavi. Dieser kann sowohl im AppNavi Widget (innerhalb Ihres Zielsystems) sowie im AppNavi Kundenportal genutzt werden. Endanwender sind vollständig anonym und benötigen keinen AppNavi Login – für sie erscheint AppNavi wenn sie sich im Zielsystem bewegen.



3 Architektur / Komponenten

AppNavi verfügt über eine Multimandantenfähige Architektur. Die oberste Einheit stellt der sogenannte Tenant dar. Innerhalb eines Tenants stehen Workspace zur Verfügung, welche wiederum Apps beinhalten.



3.1 Tenant

AppNavi verfügt über eine Tenant-Struktur. Jeder Kunde wird innerhalb eines Tenants abgebildet. Innerhalb eines Tenants bilden sogenannte Workspaces die organisatorische Struktur eines Kunden ab. Apps sind die von AppNavi unterstützten Applikationen und Contents sind die dort zur Verfügung gestellten Inhalte wie bspw. Routen oder Neuigkeiten.

3.1.1 Enterprise Authentication

Die Enterprise Authentication ermöglicht es Inhalte wie Routen, Nachrichten oder Hotspots innerhalb Ihres Unternehmens nur für authentifizierte Benutzer verfügbar zu machen. Für die Authentifikation stehen zwei verschiedene Modi zur Verfügung:

- > E-Mail-Verifizierung
- > Azure Active Directory-Authentifizierung

Bei der E-Mail-Verifizierung verifiziert sich der Nutzer mit einem Verifizierungscode, den er per E-Mail vom AppNavi-System erhält, wenn er dazu berechtigt ist. Für die AppNavi Enterprise Authentication in Verbindung mit Azure Active Directory, ist eine neue App-Registrierung in Azure Active Directory erforderlich (Typ Webanwendung).

3.1.2 Experten-Modus

Innerhalb von einzelnen Applikationen und einzelnen Schritten einer Route können Customizings in Form von JavaScript und CSS hinterlegt werden. Vor dem Hintergrund der internen IT Security kann diese Option übergeordnet im Tenant aktiviert oder deaktiviert werden.

3.1.3 Medienbibliothek

Jeder Tenant verfügt über eine Medienbibliothek. Diese kann Bilder im Format PNG, JPEG und SVG aufnehmen. Im Routenplaner hochgeladene Bilder werden in dieser Bibliothek gespeichert und erhalten einen öffentlich erreichbaren Link. Die Medienbibliothek erfüllt dabei zwei Zielsetzungen:

Zum einen ermöglicht sie es, dass in Routen verwendete Bilder für jeden User sichtbar sind. Zum anderen durchlaufen Bilder beim Hochladen in die Bibliothek einen Virensan. Dies stellt eine vertrauenswürdige Verwendung der Bilder sicher.



3.1.4 Userverwaltung

Alle Autoren innerhalb eines Tenants werden als User angelegt. Dafür wird je User eine E-Mail-Adresse, ein Passwort sowie ein Vor- und Nachname hinterlegt. Jedem User können anschließend Workspaces sowie eine Berechtigung innerhalb dieser zugeordnet werden.

Workspace Owner und Contributors sehen jeweils nur Kollegen und Inhalte ihrer eigenen Workspace(s). Damit ist die Verwendung von AppNavi innerhalb verschiedener organisatorischer Sub-Strukturen ermöglicht. Tenant Owner können alle User und alle Contents des Tenants sehen und bearbeiten.

3.2 Workspaces

Workspaces werden verwendet, um logische Strukturen innerhalb des Tenants aufbauen zu können. Je nach Kundensituation (siehe Kapitel Unterstützte Kundensituationen) gibt es verschiedene Beispiele für mögliche Strukturen:

- > Unternehmensteile
- > Standorte
- > Einzelne Projekte
- > Endkunden des AppNavi ISV
- > Etc.

Workspaces haben einen Namen und können mit Usern versehen werden. Für jeden Workspace gibt es eine Gruppe an Ownern und eine Gruppe an Contributors. Ein einzelner User kann somit unterschiedliche Berechtigungen zu verschiedenen Workspaces haben.

3.3 Apps

Um AppNavi in einem neuen System anzeigen zu können, wird innerhalb eines Workspaces eine neue App angelegt. Diese wird zunächst durch die URL des Systems definiert. Hier können sowohl absolute wie auch relative Muster angegeben werden:

- > <https://ihre-URL.de/> (AppNavi wird nur auf der Startseite angezeigt)
- > <https://ihre-URL.de/seite1/> (AppNavi wird nur auf Seite 1 angezeigt)
- > https://ihre-URL.de/* (AppNavi wird auf allen Seiten Ihres Systems angezeigt)

Apps können auch zwischen Workspaces geteilt werden. In einem solchen Szenario ist eine App immer in der Verantwortung eines Workspaces, darüber hinaus aber in anderen Workspaces sicht- und nutzbar. So können Apps gemeinsam von verschiedenen Teams mit Inhalten versehen werden – gleichzeitig ist jedoch die Verantwortlichkeit in einem Workspace verankert.

3.3.1 Einstellungen

Jede App hat eine Reihe an Einstellungen. Neben dem URL Muster einer App kann zunächst die Sichtbarkeit der App definiert werden.

Hier kann die App grundsätzlich aktiviert und deaktiviert werden – oder auch die Sichtbarkeit für einzelne Rollen innerhalb des Zielsystems festgelegt werden.

Um eine bestmögliche Unterstützung der Endanwender in der jeweiligen Zielapplikation zu gewährleisten, kann im Rahmen einer Roadmap die Verwendung von einzelnen AppNavi Features geplant werden. Hierfür lassen sich einzelne Menüpunkt (bspw. Routen oder Neuigkeiten) im AppNavi Widget aktivieren oder deaktivieren.

Für die Ermittlung der Nutzung von AppNavi Inhalten stehen Analytics Optionen zur Verfügung. Auch diese können bei Bedarf an- oder abgestellt werden.



3.3.2 Customizing Optionen

Apps können hinsichtlich ihrer Funktionalität und ihres Aussehens angepasst werden. AppNavi bietet hierfür neben einem Custom Code auch ein Custom UI Modul an.

Hinsichtlich des Aussehens können mit einfachen Color Pickern die 11 Grundfarben von AppNavi eingestellt werden. Weitere und tiefgreifendere Anpassungen lassen sich sehr einfach mit CSS anpassen.

Für funktionale Customizings steht die AppNavi Client API zur Verfügung. Hierbei handelt es sich um ein Set an Methoden und Events, welche mit Hilfe von JavaScript über AppNavi oder auch die Zielapplikation angesteuert werden können.

Sowohl das vom Autor zur Verfügung gestellte CSS, wie auch das JavaScript, werden zur Laufzeit in den Scope der Seite integriert und ausgeführt. Dies geschieht vor dem Start von AppNavi, sodass Customizings bereits greifen bevor das Widget sichtbar wird.

3.4 Contents

Mit AppNavi stehen Ihnen verschiedene Contents (Neuigkeiten, Routen, etc.) zur Verfügung, um die Nutzung Ihrer Systeme zu unterstützen. Contents sind innerhalb des Tenants zunächst einem Workspace zugeordnet. Die Zuordnung eines Contents zu einer App definiert die Bearbeitbarkeit des Contents durch die User des Tenants. Nur die einem Workspace zugeordneten User können Contents innerhalb dieses Workspaces bearbeiten.

3.4.1 Versionierung

Das Speichern eines Contents erzeugt eine neue Version des selbigen. Zu jeder Version wird das Datum der Änderung und der Name des ändernden Autors gespeichert.

3.4.2 Zuordnung zu einer App

Contents können innerhalb eines Workspaces einer oder mehreren Apps zugeordnet werden. Durch eine Zuordnung werden sie innerhalb der Zielapplikation im AppNavi Widget sichtbar.

3.4.3 Kopieren von Contents

Um auf sehr einfache Art und Weise Wiederverwendbarkeit zu erreichen, können Contents auch innerhalb von (oder zwischen) Workspaces kopiert werden.

3.4.4 Veröffentlichung in einer App

Um Contents für Endanwender sichtbar zu machen, kann die jeweils aktuelle Version des Contents veröffentlicht werden. Dies erfolgt optional auf Basis verschiedener Kriterien zur Sichtbarkeit des Contents in der jeweiligen App (Segmentierung).

Folgeversionen des Contents werden durch erneutes Veröffentlichen für die Endanwender sichtbar. Nicht veröffentlichte Zwischenversionen bleiben für die Endanwender unsichtbar.



3.5 Rollen & Berechtigungen

Um ein geeignetes Management der Inhalte zu ermöglichen, stehen den Autoren folgende Rollen zur Verfügung:

- > **Tenant Owner:** Höchste Rolle.
- > **Workspace Owner:** Kann Inhalte eines oder mehrerer Workspaces verwalten und strukturelle Änderungen am Workspace, den Apps sowie den Contents vornehmen.
- > **Workspace Contributor:** Kann innerhalb eines Workspaces Inhalte erstellen und bearbeiten.
- > **Workspace Translator:** Kann für die Übersetzung einzelner Sprachpakete innerhalb eines Workspaces berechtigt werden.

Ein User innerhalb eines Tenants kann für verschiedene Workspaces unterschiedliche Berechtigungen haben. Dies ermöglicht eine sehr einfache Mitarbeiter einzelner Autoren in verschiedenen Projekten.

4 Routen

Der zentrale Inhaltstypen sind Routen. Routen führen Ihre Nutzer entlang eines definierten Anwendungsfalls Schritt für Schritt durch die jeweilige Live-Applikation. An jeder Stelle einer Route geben Tooltips Hilfestellungen für den Nutzer, um den jeweils nächsten Schritt korrekt auszuführen. In OnBoarding-Szenarien können Routen auch kapitelweise konfektioniert und den Anwendern in Form von Sogenannten Kollektionen zur Verfügung gestellt werden.

Ihre Endanwender können Routen im AppNavi Widget unter der gleichnamigen Rubrik finden.

AppNavi stellt neben Routen weitere Inhaltstypen zur Verfügung. Je nach Kundensituation liegt der Fokus in der Nutzung von AppNavi darin zu unterstützen, zu informieren, zu beschleunigen, etc. Neuigkeiten und Kollektionen ergänzen daher das Features-Set entsprechend.

4.1 Darstellung

Innerhalb des AppNavi Widgets werden Anwendern alle für sie relevanten Routen in einem Menü angezeigt. Innerhalb dieses Menüs können Routen ausgewählt und gestartet werden.

Routen werden sortiert nach dem letzten Änderungsdatum oder basierend auf einer von den Autoren definierten Reihenfolge angezeigt. Darüber hinaus können Routen mit einer Volltextsuche gesucht und gefunden werden.

Routen haben immer im Routen-Menü immer eine Vorder- und eine Rückseite. Auf der Vorderseite ist der Titel und das Icon der Route. Auf der Rückseite befindet sich eine Beschreibung, die sichtbar wird, sobald der Endanwender mit der Maus über die Kachel einer Route fährt.

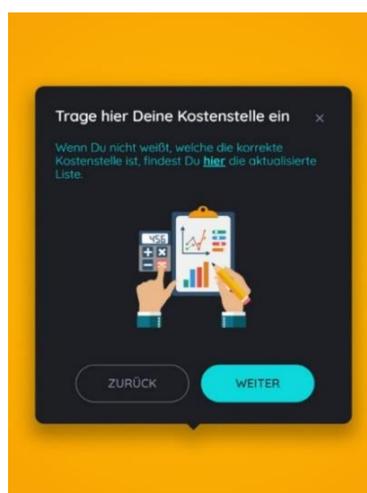
4.2 Aufbau & Bestandteile

Eine Route wird durch einen Titel, eine Beschreibung und ein Icon gekennzeichnet. Der Inhalt einer Route wird durch eine Aneinanderreihung einzelner Schritte definiert.

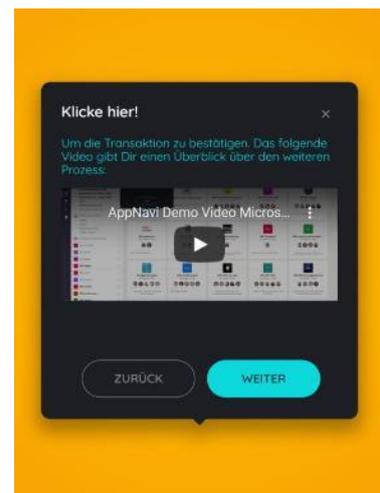
Innerhalb von AppNavi stehen Ihnen zwei Schritt-Typen zur Verfügung. Zum einen Zwischeninformationen, zum anderen an Elemente geknüpfte Informationen. Beide Schritt-Typen werden durch Tooltips repräsentiert. Diese bestehen aus einer Überschrift und einem Inhaltsbereich.



Tooltip mit Text



... mit Bildern



... mit Video



Tooltips können in Größe (Höhe und Breite) sowie Form angepasst werden. Innerhalb von Tooltips können verschiedene Inhalte eingebunden werden:

- > Text (WYSIWYG-Editor)
- > Bilder
- > Videos
- > Iframes
- > Links

Mit Hilfe des WYSIWYG-Editor können sehr einfach Formatierungen in den Tooltips vorgenommen werden. Mit Hilfe einer frei definierbaren Style-Vorlage können darüber hinaus auch einheitliche UI-Konzepte über größere AppNavi Projekte hinweg realisiert werden. Gerade bei sehr arbeitsteiligen Projekten über Themen- und Landesgrenzen hinweg ist dies ein sehr nützliches Feature.

Mit Hilfe dieser Optionen lassen sich Tooltips sehr individuell gestalten – außerdem können bspw. mit Iframes etc. auch bereits existierende Inhalte innerhalb von Routen genutzt werden.

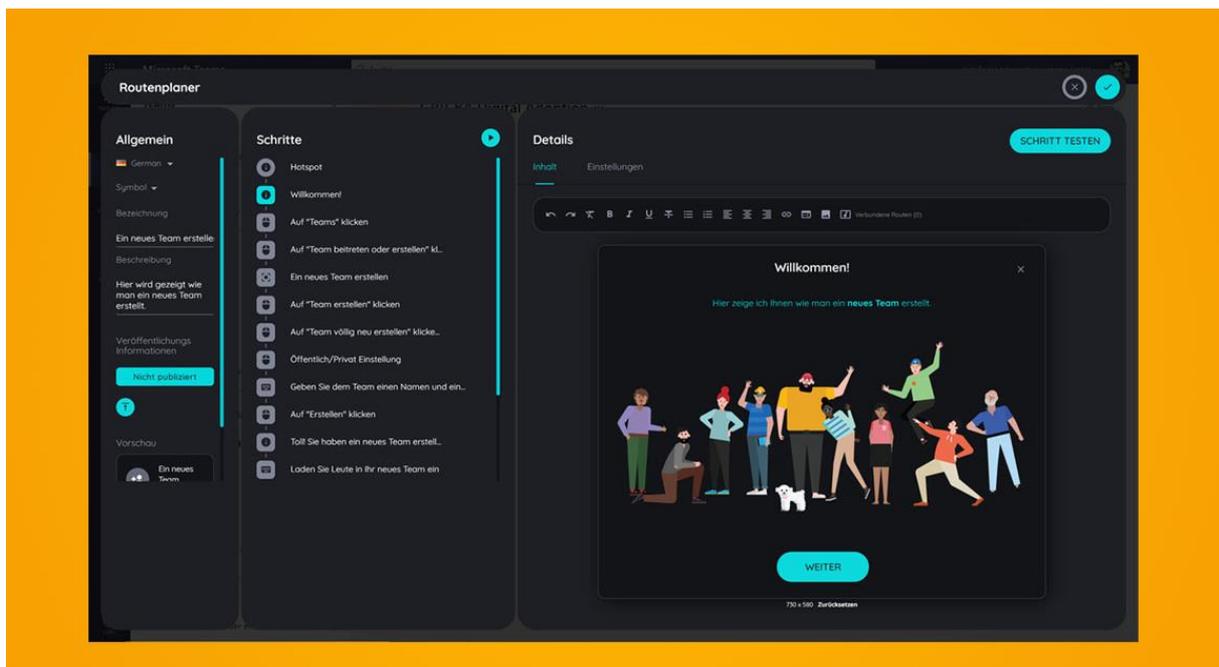
Jeder Schritt innerhalb einer Route hat einen sogenannten Interaktionstyp. Dieser legt fest, wie Endwender innerhalb der Route von einem Schritt zum nächsten zu kommen:

- > **Texteingabe** innerhalb der Applikation - Klick auf „Weiter“ im Tooltip
- > **Linksklick** innerhalb der Applikation
- > **Rechtsklick** innerhalb der Applikation
- > **Enter** innerhalb der Applikation
- > **Hover** innerhalb der Applikation
- > **Hervorheben** eines Elements innerhalb der Applikation - Klick auf „Weiter“ im Tooltip

Einzelne Schritte können mit einer Automatisierung versehen sein. Dies ermöglicht, dass der jeweilige Schritt beim Start einer Route im Speed Modus eigenständig klickt, tippt, o.ä.

4.3 Erstellen von Routen

Für die Anlage von Routen steht der Routenplaner zur Verfügung. Dieser ermöglicht es mit Hilfe einer grafischen Benutzeroberfläche sehr einfach Routen anzulegen. Technisches Verständnis für die zugrundeliegende Applikation ist hierbei nicht vonnöten.



Routenplaner



Neben der Möglichkeit, Routen mit Hilfe des Routenplaners anlegen zu können, besteht auch die Option Routen basierend auf den Aufzeichnungen im Rahmen des User Behavior Minings automatisch erstellen zu lassen.

4.4 Elementerkennung

Die Elemente zu einem Schritt werden mit Hilfe eines Fuzzy-Score-Algorithmus erkannt, welcher Elemente innerhalb einer Seite basierend auf einem statistischen Wahrscheinlichkeitsmodell erkennt. Dieses vergleicht während der Suche nach einem Schritt alle in einer Seite (DOM Struktur) vorkommenden HTML-Elemente mit einer (oder mehreren) Aufzeichnung(en) des Elements.

Als Ergebnis wird das jeweils zuverlässigste Element angezeigt, wenn dieses eindeutig in der Seite bestimmbar ist und gewisse Nebenbedingungen erfüllt – bspw. muss das identifizierte Element aktuell in der Seite sichtbar sein.

Das Besondere an diesem Verfahren ist eine hohe Zuverlässigkeit. Im Gegensatz zu marktüblichen Ansätzen speichert AppNavi keine CSS-Selektoren, sondern Kontextinformationen zu einem Element. Somit werden Elemente bspw. auch bei dynamischen Attributen und IDs sowie bspw. bei Positionsänderungen der Elemente gefunden.

Die Elementerkennung von AppNavi funktioniert sowohl innerhalb des jeweiligen HTML-Dokuments sowie in Iframes (Same Origin) und Shadow Roots innerhalb der Seite.

4.5 Trigger & Modi

Routen lassen sich über insgesamt vier verschiedene Arten starten:

- > Klick auf die **Kachel** im Routenmenü
- > Über **URL-Parameter**
- > Klick auf einen **Hotspot** (kleiner „Sticker“ an einem Element)
- > **Programmgesteuert** über die AppNavi Client API

Hinsichtlich der Modi lassen sich zwei Varianten unterscheiden. Zum einen kann eine Route im Lernmodus gestartet werden. Hier werden alle Tooltips Schritt für Schritt durchlaufen. Der Anwender klickt und tippt in der Applikation und schreitet so voran.

Darüber hinaus verfügt AppNavi über einen Speed Modus. Dieser erlaubt es, dass zuvor automatisierte Schritte automatisch durch AppNavi geklickt werden oder Text getippt wird.

4.6 Varianz & Lernen

Um Routen mit Intelligenz zu versehen, stehen verschiedene Optionen zur Verfügung. Zum einen können je Schritt verschiedene Aufzeichnungen in der HTML-Struktur der Seite aufgenommen werden. Darüber hinaus können jedoch auch einzelne Schritte angelernt werden. Dies bietet sich bspw. in Situationen an in denen der jeweilige Schritt je nach Benutzer, Sprache o.ä. leicht anders definiert ist. Autoren können hier den Schritt die verschiedenen Ausprägungen eines Elements erlernen lassen. Eine einzige Aufzeichnung ist dann ausreichend, um Varianzen in der Seite abbilden zu können.

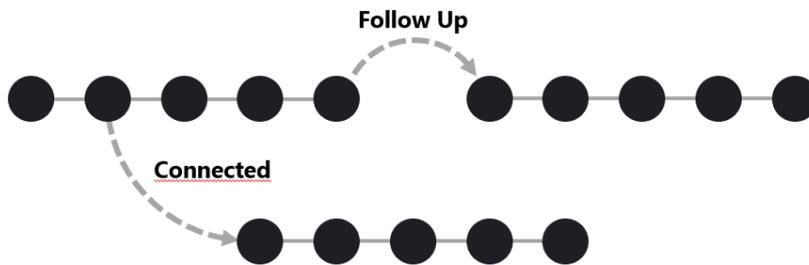
Falls einzelne Schritte innerhalb einer Route übersprungen oder anderweitig automatisch angesteuert werden sollen, kann die AppNavi Client API zum Einsatz kommen. Hier können zu gewissen Events (bspw. On Before Render) Bedingungen definiert und geprüft werden – je Ergebnis überspringt AppNavi dann einzelne Schritte oder springt in neue Routen an eine entsprechende Stelle. Dies kann vollständig automatisch erfolgen.

Gerade in komplexen Szenarien ist es darüber hinaus oftmals hilfreich, Sequenzen von Routen zu erstellen, die anschließend miteinander verbunden werden können. Dies ermöglicht es, die Anzahl der verschiedenen Routen auf ein Minimum zu reduzieren und die sich überschneidenden Teile mehrfach zu verwenden. AppNavi bietet die Möglichkeit, Routen auf zwei verschiedene Arten zu verbinden:

- > Verbinden von Routen innerhalb einer Route über "Connected Routes"
- > Verbinden von Routen jeweils am Ende über „Follow Ups“



Die folgende Grafik verdeutlicht den Unterschied zwischen den beiden Möglichkeiten:



Im Falle einer Follow Up Route werden die Endbenutzer nicht bemerken, dass sie zwei verschiedene Routen hintereinander bearbeiten. Die Endbenutzer müssen auch keine Folgeroute auswählen, sondern sie wird einfach gestartet. Es ist außerdem möglich, Routen aus verschiedenen Anwendungen auszuwählen, so dass Sie Anwendungsfälle auch Applikationsübergreifend nahtlos aneinander anschließen können.

4.7 Automatisierung & RPA

Um die KPIs „Steigerung der First-Time-Right“ und „Steigerung der Effizienz“ zu erhöhen, verfügt AppNavi über eine Funktionalität für **Attended RPA**. Hier steht der Nutzer und seine Geschwindigkeit in der Bearbeitung seiner Aufgaben im jeweiligen System im Fokus.

Innerhalb einer Route können einzelne oder auch alle Schritte einer Route automatisiert zur Verfügung stehen. Informationsschritte werden nicht automatisiert.

Je nach Startmodus einer Route werden diese Schritte dann im Lern- oder im Geschwindigkeitsmodus durchlaufen. Der für die jeweilige Route gültige Standardmodus kann mit Hilfe des Routenplaners eingestellt werden – der jeweilige Alternativ-Modus ist für Endanwender dann jeweils innerhalb der Routenkachel erreichbar.

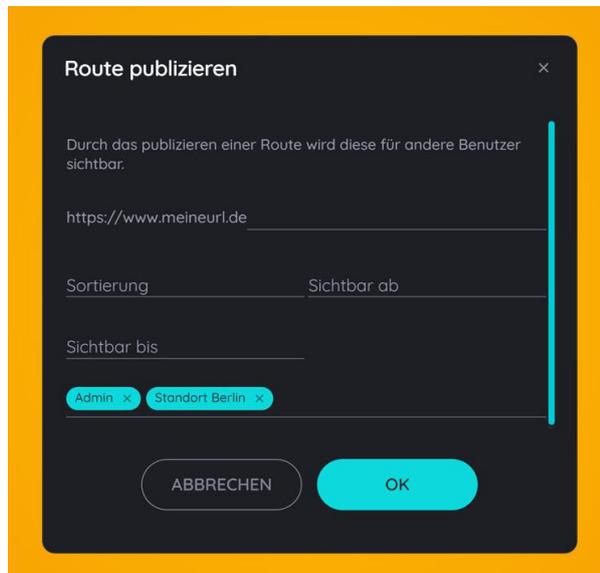
Auch wenn Routen über die API gestartet werden, stehen beide Modi zur Verfügung.

4.8 Versionierung & Publikation

AppNavi verfügt über einen zweistufigen Publikations-Mechanismus. Eine von einem Autor erstellte Route kann zunächst gespeichert und damit den anderen Autoren zur Verfügung gestellt werden. Jedes Speichern erzeugt dabei eine neue Version der Route.



Sobald ein Bearbeitungsstand eine gewisse Reife erreicht hat, kann die Route publiziert werden. Dies macht sie für die Anwender in der aktuellen Version sichtbar. Im Hintergrund können die Autoren mit der Bearbeitung fortfahren und neue Arbeitsversionen erzeugen, welche mit einer erneuten Publizierung den Anwendern zugänglich gemacht werden können.



4.9 Sichtbarkeit und Segmentierung

Die Publikation einer Route (gleiches gilt auch für alle anderen Contents) kann mit Regeln verknüpft sein. Routen können somit an gewisse Segmente der Endanwender ausgeliefert werden.

Aufgrund seiner Datenschutz-freundlichen Architektur, liefert AppNavi beim Aufruf einer Seite durch den Endanwender zunächst alle AppNavi Inhalt der Seite aus – im Client wird anschließend entschieden, welche der Inhalte ein- bzw. ausgeblendet werden. Dies ermöglicht eine Zielgenaue Auslieferung der Inhalte ohne Informationen zum jeweiligen User zu benötigen.

Typische Segmentierungen:

- > Nach Benutzerrolle
- > Nach Standort
- > Nach Land

Für die clientseitige Segmentierung können in der Seite befindliche Kriterien wie auch System-externe Informationen (bspw. per Client API) genutzt werden.

Für eine einfache und über einen Tenant hinweg wiederverwendbare Segmentierung können Target Audiences gebildet werden. Dies sind Regelsätze (bspw. HR Business Partner / Region Süd), welche eine zielgenaue Ausspielung sehr einfach möglich machen.

4.10 Weitere Features

Für die Routen innerhalb von AppNavi stehen eine Reihe an weiteren Features zur Verfügung. Hier eine Auswahl:

Mehrsprachigkeit: Routen können in verschiedene Sprachen übersetzt werden. Dies ist sowohl im Routenplaner wie auch im Kundenportal möglich. Neben den jeweiligen Sprachen können auch Dialekte als eigenes Paket angeboten werden. Die Aufzeichnung einer Route erfolgt dabei über alle Sprachen hinweg jeweils nur einmal – d.h. bei Änderungen schlägt sich eine einzige Anpassung automatisch in allen Sprachen nieder.

Automatische Übersetzung: Insgesamt 29 Sprachen können (ohne zusätzliche Kosten) mit Hilfe einer DeepL Integration automatisch übersetzt werden.



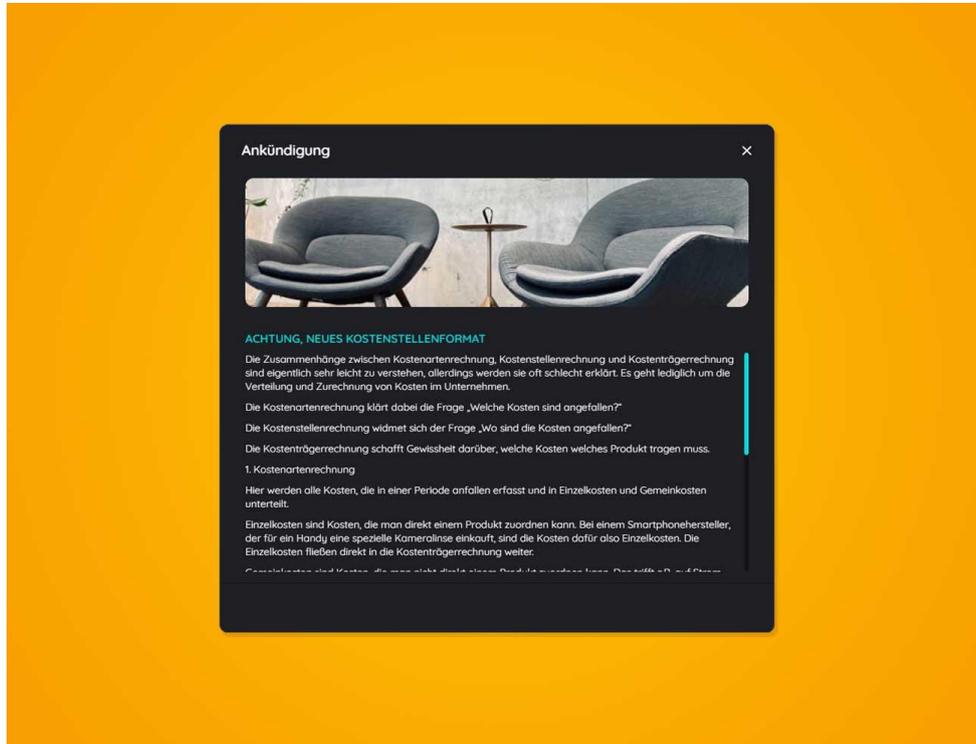
Responsiveness: AppNavi erkennt die Auflösung der Applikation und passt die Optik entsprechend an – auch bei sich ändernden Fenstergrößen.

Auto Scrolling & Tracking: Wenn das nächste Element nicht im sichtbaren Bereich ist, scrollt unser Algorithmus an die richtige Stelle: Horizontal und vertikal, in der Seite sowie in Iframes.

5 Neuigkeiten & Ankündigungen

Neuigkeiten erlauben es, für Anwender kleine Artikel zielgerichtet anzeigen zu können. Auch hier greift die Segmentierung, sodass die Artikel nur den relevanten Anwendern und auch nur an der relevanten Stelle im System angezeigt wird.

Neuigkeiten können sowohl aus dem AppNavi Widget heraus, wie auch als Ankündigung oder basierend auf selbst definierten Triggern (bspw. beim erstmaligen Besuch der Seite initiiert) werden.



Genau wie alle anderen Inhalte von AppNavi auch, können Neuigkeiten einmal zentral innerhalb eines Workspaces angelegt und dann in verschiedenen Apps genutzt werden. Dies ermöglicht eine zentrale Anlage und Pflege, aber gleichzeitig eine dezentrale Nutzung von Neuigkeiten.

5.1 Darstellung

Innerhalb des AppNavi Widgets befindet sich ein eigener Menüpunkt für Neuigkeiten. Eine Liste mit allen (der jeweiligen Segmentierung entsprechenden) Neuigkeiten wird angezeigt, sobald dieser Menüpunkt geöffnet wurde.

Neuigkeiten werden durch ein Bild, einen Titel und eine Beschreibung in der Liste dargestellt. Klickt der Nutzer eine Neuigkeit an, öffnet sich ein Detailbereich, welcher die Inhalte anzeigt.

5.2 Aufbau & Bestandteile

Eine Neuigkeit besteht aus einem Hero Bild und dem eigentlichen Inhalt der Neuigkeit. Dieser kann ebenfalls mit einem WYSIWYG-Editor bearbeitet werden:

- > Text (WYSIWYG-Editor)
- > Bilder
- > Videos
- > Iframes
- > Links



5.3 Trigger & Modi

Neuigkeiten lassen sich über insgesamt drei verschiedene Arten starten:

- > Klick auf die **Kachel** im Menü
- > Klick auf einen **Hotspot** (kleiner „Sticker“ an einem Element)
- > **Programmgesteuert** über die AppNavi Client API



6 Hotspots

Hotspots ermöglichen es, jede browserbasierte Software mit zusätzlichen Informationen anzureichern, die den Nutzer bei der optimalen Nutzung der Anwendung unterstützen. So können Elemente kurzzeitig oder dauerhaft mit relevanten Anwendungs- oder Prozessinformationen ergänzt werden. Folgende Typen von Hotspots lassen sich unterscheiden:

- > **Hinweise anzeigen:** Ein einzelner Tooltip wird beim Klicken auf den Hotspot angezeigt
- > **Ankündigungen einblenden:** Eine Ankündigung wird geöffnet
- > **Routen starten:** Eine Route in einer zu wählenden App wird an einem wählbaren Schritt gestartet

Um die Erstellung und Pflege von Hotspots so einfach wie möglich zu gestalten, ist das Design weitgehend der Erstellung von Routen angelehnt.

6.1 Darstellung

Hotspots können als kleine runde Kreise innerhalb des UIs der Zielapplikation hinterlegt werden. Je nach Typ haben sie ein entsprechendes Symbol.

6.2 Trigger & Modi

Hotspots werden immer selbständig im UI des Zielsystems gesucht und eingezeichnet, sobald diese gefunden wurden. Per Client API können Hotspots ein- und ausgeblendet werden.

Klickt ein Nutzer auf einen Hotspot, wird dieser ausgelöst und je nach Typ eine entsprechende Aktion ausgelöst.



7 Trainings & Kollektionen

Als Feature aus dem Bereich OnBoarding können Kollektionen verwendet werden. Kollektionen erlauben es, Routen zu Kapiteln zusammenzufassen. Eine Kollektion kann wiederum einer App zugeordnet und anschließend publiziert werden. Anwender können Routen innerhalb der Kollektionen durchlaufen und den Fortschritt sehen. Sobald alle Routen einer Kollektion durchlaufen sind, wird die Kollektion als abgeschlossen markiert.

Wenn eine Kollektion einer App zugeordnet wird, werden alle darin befindlichen Routen ebenfalls dieser App zugeordnet (falls dies nicht bereits der Fall ist). Wenn eine Kollektion im umgekehrten Fall von einer App entfernt wird, verbleiben die Routen zugeordnet.

Wenn eine Kollektion gelöscht wird, kann der User angeben, ob (alle) Routen der Kollektion ebenfalls gelöscht werden sollen. Wenn sie gelöscht werden, werden sie vollständig gelöscht - mit all ihren weiteren Zuordnungen und Publizierungen in etwaigen anderen Apps. Wenn der Nutzer dies verneint, bleiben die Routen in den jeweiligen Apps mit unveränderten Zuordnungen und Publizierungen verfügbar. Die Kollektion selbst wird anschließend in beiden Fällen gelöscht (inkl. Zuordnungen und Publizierungen).

Wenn eine Kollektion veröffentlicht wird, werden alle in der Kollektion befindlichen Routen ebenfalls veröffentlicht. Hier lassen sich die folgenden Fälle unterscheiden:

1. Eine Route der Kollektion ist bereits veröffentlicht
2. Eine Route der Kollektion ist bereits veröffentlicht, aber aktuell unsichtbar
3. Eine Route ist noch nicht veröffentlicht
4. Eine Route wird (während sie in einer Kollektion veröffentlicht ist) in der App depubliziert
5. Die Kollektion wird depubliziert

Zu 1: Die Route erscheint im Routenmenü und in der Kollektion

Zu 2, 3, 4: Die Route bleibt im Routenmenü unsichtbar, erscheint aber in der Kollektion

Zu 5: Die Kollektion wird unsichtbar. Routen die vorher im Menü sichtbar waren, bleiben dies auch - alle unsichtbaren Routen bleiben unsichtbar



8 Analytics

AppNavi stellt Endanwendern bei der Nutzung neuer oder bestehender Systeme verschiedene Hilfearten zur Verfügung. Um den Autoren dieser Hilfen einen besseren Einblick in die Verwendung ihrer Inhalte geben zu können und dadurch die Qualität der Hilfestellung zu verbessern, stellt AppNavi ein ausgereiftes Analytics Modul zur Verfügung.

Hier werden auf drei Ebenen Analysedaten gesammelt und innerhalb des AppNavi Portals mit Hilfe von Dashboards zur Anzeige gebracht:

- > Content-Ebene (Route, Neuigkeit, etc.)
- > App-Ebene
- > Workspace-Ebene

8.1 Datenerfassung

Um Analysedaten anzeigen zu können, sammelt AppNavi verschiedene Nutzerdaten. Dafür muss die Analysefunktion aktiviert sein. Standardmäßig werden die folgenden Benutzerdaten gesammelt:

- > **Browser-Name**
- > **Sprache des Benutzers**
- > **Session ID:** Die Session ID wird vom System für jede Benutzersitzung generiert. Eine Benutzersitzung beginnt, wenn eine Anwendung im Browser geöffnet wird und endet, wenn der Benutzer die Anwendung z. B. durch Schließen der Registerkarte oder des Browsers schließt.
- > **User ID:** Standardmäßig wird von AppNavi für jeden Benutzer eine User ID generiert und im lokalen Speicher des Browsers des Benutzers gespeichert. Die Benutzerkennung kann jedoch auch durch Custom Code festgelegt werden, um bspw. OKRs oder ähnliches zu erheben.

8.2 Metriken

In AppNavi Analytics werden verschiedene Metriken dargestellt, um das Nutzerverhalten über verschiedene Inhalte hinweg aufzuzeigen. Die grundlegenden Metriken werden im Folgenden beschrieben.

| Aktive Benutzer

Ein aktiver Benutzer ist ein Benutzer, der eine Anwendung mindestens einmal in einem bestimmten Zeitraum aufgerufen hat. Wenn ein Benutzer mit einer eindeutigen Kennung (User ID) eine Anwendung in einem bestimmten Zeitraum mehrmals aufgerufen hat, wird der Benutzer nur als ein Benutzer gezählt.

| Interagierende Benutzer

Ein interagierender Benutzer ist ein eindeutiger Benutzer, der mindestens einmal mit AppNavi interagiert hat. Eine Interaktion könnte sein:

- > Start einer Route
- > Fortsetzen einer Route
- > Lesen einer Neuigkeit oder Ankündigung
- > Klick auf den Avatar
- > Initiieren eines Hotspots

| Wiederkehrende Benutzer

Ein wiederkehrender Benutzer ist ein Benutzer, der AppNavi innerhalb eines bestimmten Zeitraums mindestens zweimal verwendet hat.



8.3 Diagramme

AppNavi verwendet deskriptive Diagramme zur Visualisierung von Analysen. Die verschiedenen Diagrammtypen werden im Folgenden beschrieben.

8.3.1 Allgemein

AppNavi verwendet ein hierarchisches Analysemodell: Ausgehend von der globalen Ebene (alle Daten eines Workspaces innerhalb eines Tenants) können die Daten granularer betrachtet werden. Dies beinhaltet Einblicke in Daten auf Anwendungsebene, aber auch Daten auf Inhaltsebene innerhalb einer Anwendung.

- > **Dashboard auf der Startseite**
Nach einem Klick auf den Menüpunkt "Home" im AppNavi-Menü des Kundenportals erscheint das Dashboard. Dieses zeigt alle innerhalb eines Tenants aggregierten Daten und das darin ausgewählte Workspaces an.
- > **Dashboard zu einer App**
Die im Dashboard angezeigten Charts können auch auf Applikationsebene angezeigt werden.
- > **Dashboard zu einem Inhaltselement**
Die im Dashboard und auf Applikationsebene angezeigten Diagramme können auch auf der Ebene der Inhaltselemente (Routen, Nachrichten etc.) angezeigt werden.

Im Folgenden werden die einzelnen Standard-Diagrammtypen beschrieben. Bei den meisten Diagrammen kann der Zeitraum ausgewählt werden:

- > **Letzte 30 Tage:** Hier werden die Daten der letzten 30 Tage angezeigt.
- > **Gesamt:** Diese Einstellung visualisiert alle Daten seit der Erstellung des Workspaces, der Anwendung oder des jeweiligen Inhaltselements auf Monatsebene

8.3.2 KPIs zum Workspace

Kleine Kacheln innerhalb des Dashboards zeigen KPIs und deren Veränderungen innerhalb der letzten 30 Tage an.

- > **Aktive Nutzer**
Diese Kennzahl zeigt die Anzahl der aktiven Nutzer an. Aktive Nutzer sind Nutzer, die die zugrundeliegende Anwendung im gewählten Zeitintervall genutzt haben. Zusätzlich wird die Veränderung dieser Kennzahl über die letzten 30 Tage als prozentuale Veränderung dargestellt.
- > **Interagierende Nutzer**
Diese Zahl zeigt alle Nutzer, die AppNavi genutzt haben und die Veränderung dieser Kennzahl.
- > **Verwendete Zeit**
Diese Zahl zeigt die Zeit aller interagierenden Nutzer von AppNavi und die Veränderung dieser Kennzahl.

Die Verwendete Zeit wird auf der Grundlage der Interaktionszeit berechnet. Folgende Zeiten fließen in die Berechnung mit ein:

- > Die Dauer eines Routendurchlaufs (Dauer aller Schrittanzeigen innerhalb einer Route).
- > Die Anzeigedauer einer Neuigkeit oder einer Ankündigung
- > Die Anzeigedauer eines Hinweises mit Hilfe eines Hotspots

8.3.3 Benutzeraktivität

| Inhalt

Dieses Liniendiagramm stellt den Verlauf dreier verschiedener Nutzertypen dar:



> Aktive Nutzer

Diese Linie zeigt die aktiven Nutzer im ausgewählten Zeitintervall. Aktive Benutzer sind Nutzer, die die zugrundeliegende Anwendung in dem ausgewählten Zeitintervall verwendet haben.

> Interagierende Nutzer

Diese Linie zeigt alle Nutzer, die AppNavi benutzt haben.

> Wiederkehrende Nutzer

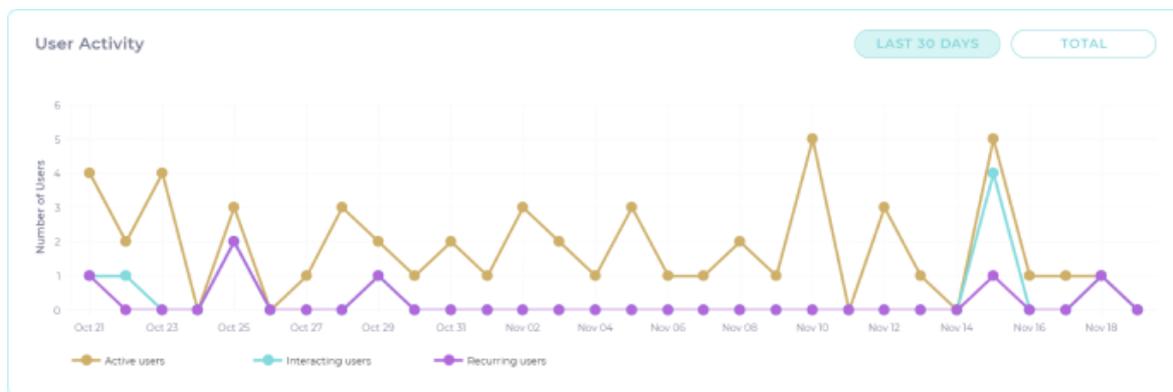
In dieser Linie werden Nutzer angezeigt, die im ausgewählten Zeitraum mehrfach aktiv waren.

| Optionen

- > Linie ein- oder ausblenden
- > Zeitraum wählen

| Hinweis

Wenn eine Anwendung deaktiviert ist oder nicht mehr im Tenant bzw. Workspace existiert, werden die Daten nicht in das Diagramm aufgenommen.



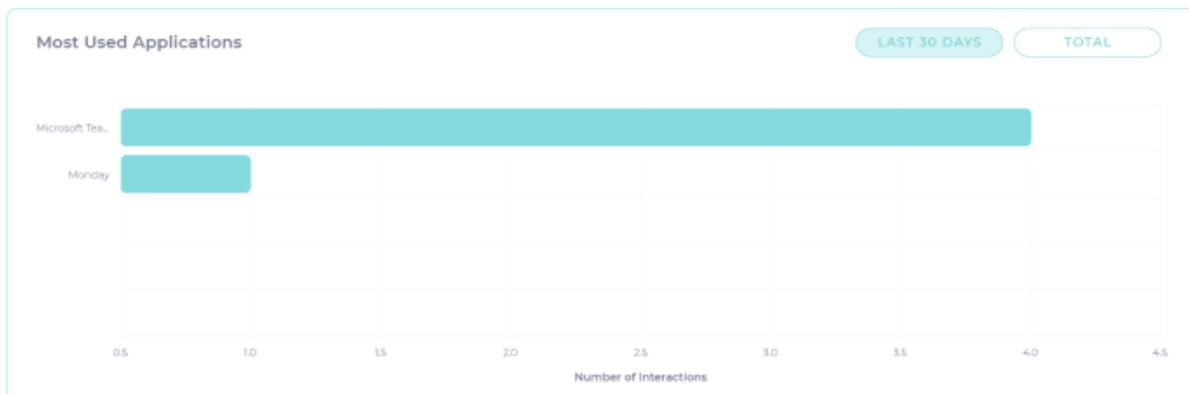
8.3.4 Meistgenutzten Anwendungen

| Inhalt

Dieses Balkendiagramm zeigt die Top 5 der am häufigsten genutzten Anwendungen innerhalb des Workspaces. Die Rangliste basiert auf den interagierenden Nutzern pro Anwendung. Es werden alle aktiven und nicht gelöschten Anwendungen innerhalb des Workspaces berücksichtigt.

| Hinweis

Wenn Sie auf den Titel eines Balkens klicken, können Sie detaillierte Daten zu der ausgewählten Anwendung anzeigen.





8.3.5 Inhaltsverwendung

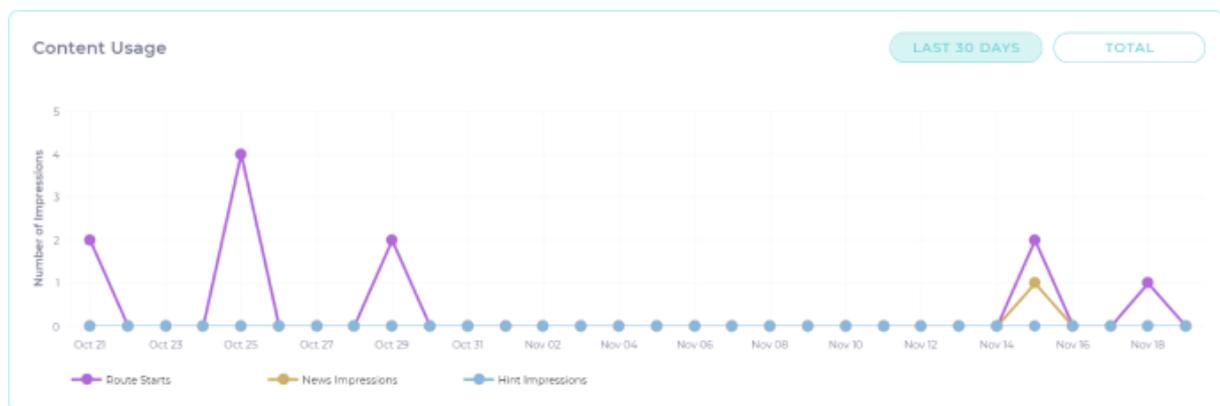
Inhalt

Dieses Liniendiagramm zeigt die Nutzung von drei verschiedenen Inhaltstypen:

- > **Starts von Routen**
Die Linie zeigt die Häufigkeit an, mit der Routen im ausgewählten Zeitraum gestartet wurden. Gezählt werden auch die Routen, die über Hotspots gestartet wurden.
- > **Nachrichten Impressions**
Die Zeile zeigt die Häufigkeit an, mit der im ausgewählten Zeitraum Nachrichten gestartet oder Ankündigungen angezeigt wurden. Gezählt werden auch die Nachrichten oder Ankündigungen, die über Hotspots initiiert wurden.
- > **Hinweisaufrufe**
Die Zeile zeigt die Häufigkeit an, mit der im gewählten Zeitraum Hinweise über Hotspots initiiert wurden.

Optionen

- > Linie ein- oder ausblenden
- > Zeitraum wählen



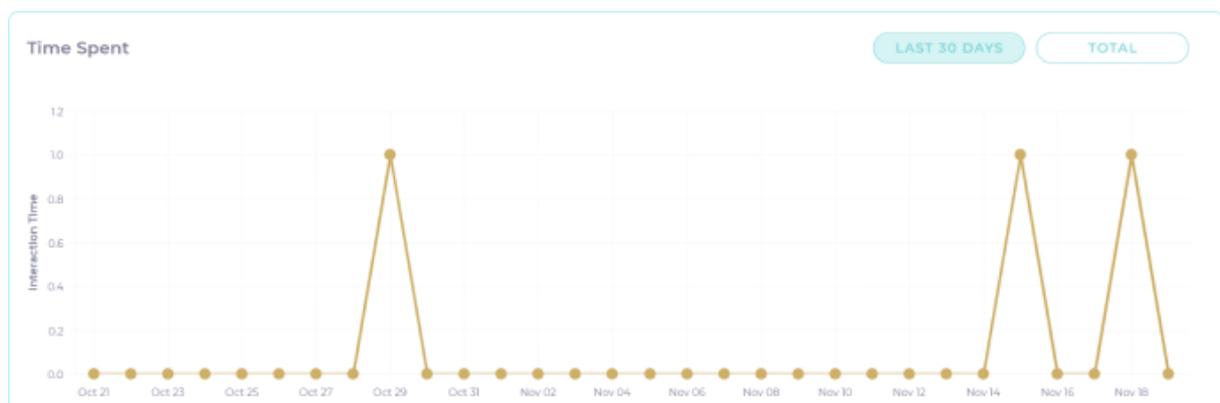
8.3.6 Aufgewendete Zeit

Inhalt

Die Linie zeigt an, wie viel Zeit die Nutzer im ausgewählten Zeitraum mit Inhalten von AppNavi verbracht haben.

Optionen

- > Zeitraum wählen





8.3.7 Übersicht der Anwendungen

Inhalt

Die Tabelle zeigt alle von AppNavi unterstützten Anwendungen innerhalb des gewählten Workspaces an. Folgenden Informationen lassen sich der Tabelle entnehmen.

- > **Name:** Titel der Anwendung
- > **Features:** Verwendete Features in der jeweiligen Anwendung (Routen, Nachrichten, Kollektionen, Hotspots)
- > **Routen:** Anzahl der Routen in der jeweiligen Anwendung
- > **Inhalte:** Anzahl der Inhalte (Nachrichten, Ankündigungen, Hinweise) in der jeweiligen Anwendung
- > **Analytics:** Zeigt an, ob die Analytics aktiviert oder deaktiviert sind.
- > **Theme:** Zeigt an, ob das dunkle Standardtheme oder ein individuelles Theme verwendet wird
- > **Aktiv:** Zeigt an, ob AppNavi in der jeweiligen Anwendung aktiviert oder deaktiviert ist

Optionen

- > Nach Stichwort suchen
- > Neue App hinzufügen
- > Neu laden

NAME	FEATURES	ROUTES	CONTENTS	ANALYTICS	THEME	ACTIVE
AppNavi Learning Center	📄 📍	3	2	🟢	dark	🟢
AppNavi Portal	📄 📍	0	0	🟢	dark	🟢
AppNavi Test	📄 📍	0	0	🟢	dark	🟢
🔗	📄 📍	7	1	🟢	dark	🔴
🔗	📄 📍	2	3	🟢	dark	🟢

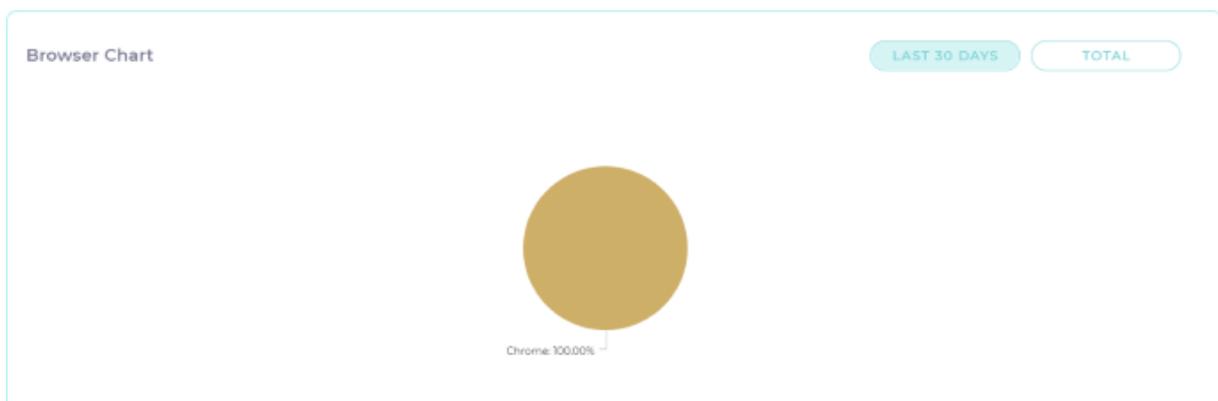
8.3.8 Verwendete Browser

Inhalt

Das Kreisdiagramm zeigt den prozentualen Anteil der Nutzung der Anwendung durch verschiedene Browser.

Optionen

- > Zeitraum wählen





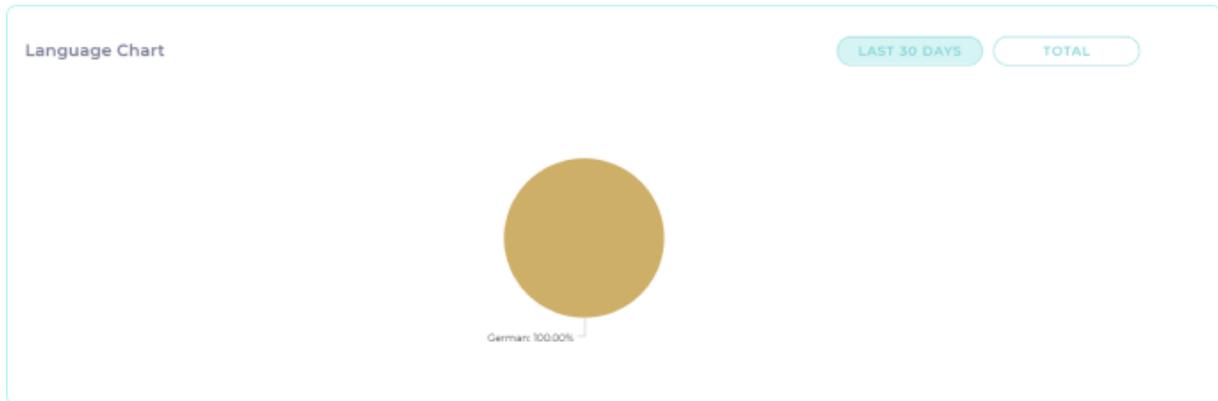
8.3.9 *Verwendete Sprachen*

| Inhalt

Das Kreisdiagramm zeigt die prozentuale Nutzung der Anwendung in den jeweiligen Browser-Sprachen.

| Optionen

- > Zeitraum wählen



8.3.10 *Routenprobleme*

| Inhalt

Diese Tabelle enthält Informationen über mögliche Beeinträchtigungen von Routen. Autoren können sich damit einen schnellen Überblick verschaffen, welche Routen möglicherweise überprüft werden müssen. Folgende Informationen sind in der Tabelle ersichtlich:

- > **Routentitel:** Titel der jeweiligen Strecke
- > **Schritttitle:** Titel des Schrittes, der möglicherweise untersucht werden muss
- > **Fehlerquote:** Berechnete Fehlerquote des entsprechenden Schrittes
- > **Kritikalität:** Ermittelte Kritikalität des potentiellen Fehlers

| Optionen

- > Nach Stichwort suchen
- > Neu laden

Route issues (last 30 days)

Search route issues

RELOAD

ROUTE TITLE	STEP TITLE	ERROR DATE	CRITICALITY
No data available			

8.3.11 *KPIs zu einer Route*

Kleine Kacheln innerhalb des Dashboards zeigen KPIs innerhalb der letzten 30 Tage an.

- > **Routenstarts**

Diese Kennzahl zeigt, wie oft die ausgewählte Route angezeigt wurde und wie sich die Zahl in den letzten 30 Tagen verändert hat.



> Einmalige Routenstarts

Diese Zahl zeigt an, wie oft die gewählte Route von verschiedenen Nutzern angezeigt wurde und wie sich die Zahl in den letzten 30 Tagen verändert hat.

> Durchschnittliche Anzeigedauer

Diese Zahl zeigt an, wie lange einzelne Schritte der Route im Durchschnitt angezeigt wurden und wie sich die Zahl in den letzten 30 Tagen verändert hat.

8.3.12 Routenfortschritt

| Inhalt

Diese Tabelle enthält Informationen über Schritte in einer Route und deren Erreichung im Verlauf der Routennutzung.

- > **Schritttitle:** Titel des Routenschritts.
- > **Anzeigedauer (Sekunden):** Anzahl der Sekunden, die ein Schritt im Durchschnitt angezeigt wird.
- > **Vom Benutzer angezeigt:** Prozentsatz der Anzeige dieses Schrittes über alle Routendurchläufe.

| Optionen

- > Nach Stichwort suchen
- > Neu laden

Steps	SEARCH	LAST 30 DAYS	TOTAL
STEP TITLE	DISPLAY DURATION (SECONDS)	SHOWN BY USER	
Willkommen!	4	100%	
Schritt 1: Klicken Sie auf 'Chat'	4	75%	
Schritt 2: Klicken Sie auf das Bearbeitungsymbol	32	75%	
Schritt 3: Geben Sie einen Namen ein	6	10%	
Schritt 4: Schreiben Sie eine Nachricht	0	0%	
Schritt 5: Schließen Sie Ihre Nachricht los	0	0%	
Herzlichen Glückwunsch!	0	0%	

8.3.13 KPIs zu einer News bzw. einer Ankündigung

Kleine Kacheln innerhalb des Dashboards zeigen KPIs und deren Veränderungen innerhalb der letzten 30 Tage an.

> Aufrufe

Diese Kennzahl zeigt, wie oft die ausgewählte Neuigkeit oder Ankündigung angezeigt wurde und wie sich die Zahl in den letzten 30 Tagen verändert hat.

> Eindeutige Aufrufe

Diese Zahl zeigt an, wie oft die ausgewählte Neuigkeit oder Ankündigung von verschiedenen Nutzern angezeigt wurde und wie sich die Zahl in den letzten 30 Tagen verändert hat.

> Durchschnittliche Anzeigedauer

Diese Zahl gibt an, wie lange die Neuigkeiten oder Ankündigungen im Durchschnitt angezeigt wurden und wie sich der Wert in den letzten 30 Tagen verändert hat.

8.3.14 Neuigkeiten

| Inhalt

Dieses Liniendiagramm zeigt die Anzahl der Aufrufe von Neuigkeit oder Ankündigungen im ausgewählten Zeitraum:

> Aufrufe

Diese Zeile zeigt alle Aufrufe von Neuigkeiten und Ankündigungen durch alle Benutzer.

> Eindeutige Impressionen

Diese Zeile zeigt alle Aufrufe von Neuigkeiten und Ankündigungen durch einzelne Nutzer an.

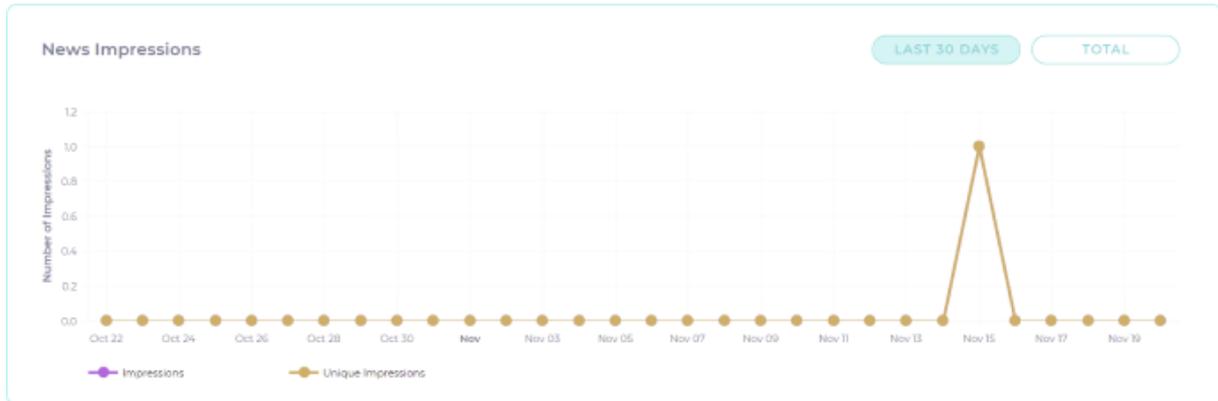


Optionen

- > Zeilen ein- oder ausblenden
- > Zeitraum wählen

Hinweis

Wenn eine Anwendung deaktiviert ist oder im Tenant oder Workspace nicht mehr existiert, werden die Daten nicht in das Diagramm aufgenommen.





9 User Behavior Mining

Mit Hilfe von User Behavior Mining ist es möglich, Informationen über das Verhalten von Nutzern in einer Anwendung zu erhalten. Die Auswertungen des User Behavior Mining liefern beispielsweise Antworten auf folgende Fragen:

- > Wie viele Nutzer verwenden die Software?
- > Wie lange wird die Software von einem Nutzer genutzt?
- > Wie regelmäßig und zu welchen Zeiten wird die Software genutzt?
- > Welche Funktionen und Prozesse einer Software werden genutzt?
- > An welchen Stellen in der Software treten Probleme auf?

User Behavior Mining ist mit additiven Kosten versehen. Vor einem Einsatz müssen Sie sicherstellen, dass die Funktion freigeschaltet ist.

9.1 Verwalten von User Behavior Mining-Kampagnen

User Behavior Mining Kampagnen werden über den Kontext der Anwendung in der Anwendungsübersicht verwaltet. Voraussetzung dafür ist, dass der Anwendung ein Vertrag mit aktiviertem User Behavior Mining zugewiesen ist.

Die Aufzeichnung von Analysedaten erfolgt immer in sogenannten Kampagnen. Eine Kampagne hat eine feste Dauer von 30 Tagen. Das Startdatum der Kampagne kann definiert werden. Es kann immer exakt eine aktive Kampagne pro Anwendung für eine bestimmte Zeitspanne geben. Kampagnen können jederzeit angehalten und wieder fortgesetzt werden. Wenn eine Kampagne pausiert wird, werden keine Daten zum Benutzerverhalten gesammelt.

9.2 Interaktions-Objekte

Ein Interaktionsobjekt ist ein UI-Element der Webanwendung, mit dem der Benutzer interagiert hat. Unter Interaktionen verstehen wir:

- > Linksklick
- > Rechtsklick
- > Tastatureingabe

Das Interaktionsobjekt wird bei der ersten Interaktion erfasst. Allerdings werden Interaktionsobjekte immer nur einmal pro Anwendung erfasst. Eine Aktion, wie z.B. das Ändern des Titels eines Interaktionsobjekts, wirkt sich daher immer auf alle Kampagnen innerhalb einer Anwendung aus. Die folgenden Informationen werden in der Übersicht der UI-Interaktionselemente angezeigt:

- > **Titel:** Der erfasste oder geänderte Titel eines Interaktionselements.
- > **Seiten-Url:** Die URL der Seite, auf der das Interaktionselement erfasst wurde.
- > **Letzte Entdeckung:** Der Zeitpunkt, zu dem das Element erfasst wurde.
- > **Wiedererfassung:** Zeigt an, ob sich das Element im Wiederauffindungsmodus befindet.
- > **Aktiv:** Zeigt an, ob das Element aktiv ist oder nicht.

| Wiederauffindung

Jedes UI-Interaktionselement enthält beschreibende Informationen über das Element, die zum Auffinden eines Elements verwendet werden. Wenn diese Informationen aktualisiert werden müssen (z. B. weil ein Element nicht mehr gefunden werden kann), kann der Wiederauffindungsmodus aktiviert werden. Wenn ein Benutzer ein Ereignis auf dem Element auslöst, werden die aktualisierten Informationen gespeichert.

| Aktivieren und Deaktivieren

Wird ein Element deaktiviert, wird es im Beschriftungsassistenten nicht mehr gesucht und angezeigt.



9.3 Sessions

Sessions bieten die Möglichkeit, das Verhalten einzelner Benutzer in einer Sitzung zu analysieren. Auf diese Weise kann ein Gefühl dafür entwickelt werden, wie sich Benutzer in einer Anwendung verhalten. Eine Sitzung beginnt, wenn ein Browserfenster oder der Tab geöffnet wird und endet, wenn das Browserfenster oder der Tab geschlossen wird. Alle aufgezeichneten Daten sind anonym. Es ist daher möglich festzustellen, wie sich ein Benutzer in einer Anwendung verhält, aber nicht, wer dieser Benutzer ist. Die Ansicht Sitzungsdetails zeigt den genauen Ablauf einer Benutzersitzung.

9.4 Behavior Flows

Ein Behavior Flow ist ein Verhaltensmuster, das durch einen KI-gestützten Mechanismus identifiziert wurde. Dazu wird das Verhalten aller Nutzer einer Kampagne analysiert und die Verhaltensmuster mit der höchsten Ausführungshäufigkeit angezeigt. Innerhalb des Profils einer Kampagne können mit Hilfe eines Dashboards (Behavior Flow Cockpits) die folgenden Informationen angezeigt werden:

- > Die Anzahl der vollständigen Prozessausführungen
- > Die Anzahl der Teilprozessausführungen
- > Die durchschnittliche Zykluszeit des Prozesses
- > Die vollständigen und teilweisen Prozessausführungen im Zeitverlauf
- > Ein Heatmap-Diagramm zeigt den Verlauf der Prozessausführungen nach Wochentagen und Zeitfenstern
- > Darstellung der einzelnen Aktionen innerhalb des Verhaltensmusters. Der Wert Time To Reach zeigt, wie viel Zeit im Durchschnitt von der Ausführung eines Schrittes bis zur Ausführung des nächsten Schrittes vergangen ist.

9.5 Automatisierte Erstellung von Routen

Die meisten gängigen Verhaltensweisen von Nutzern können mit Hilfe von Verhaltensabläufen in eine Route umgewandelt werden. D.h. der Nutzer kann Routen auf der Grundlage der in einer Kampagne verfügbaren Verhaltensabläufe erstellen.

9.6 Datensicherheit

Die Daten, die während einer aktivierten User Behavior Mining-Kampagne gesammelt werden, sind nicht personenbezogen. Die folgenden Analytics-Daten werden gesammelt:

- > Browser
- > Sprache
- > Aktion (Linksklick / Rechtsklick / Texteingabe)
- > Element-Informationen

Es werden keine Daten von der Seite selbst gesammelt, wie zum Beispiel der Text einer Benutzereingabe. Hierfür wird eine **Data Clearing Algorithmus** verwendet, welcher Namen, Telefonnummern, Bankverbindungen etc. entfernt.

10 Möglichkeiten für Customizing

10.1 Standard Text Styling

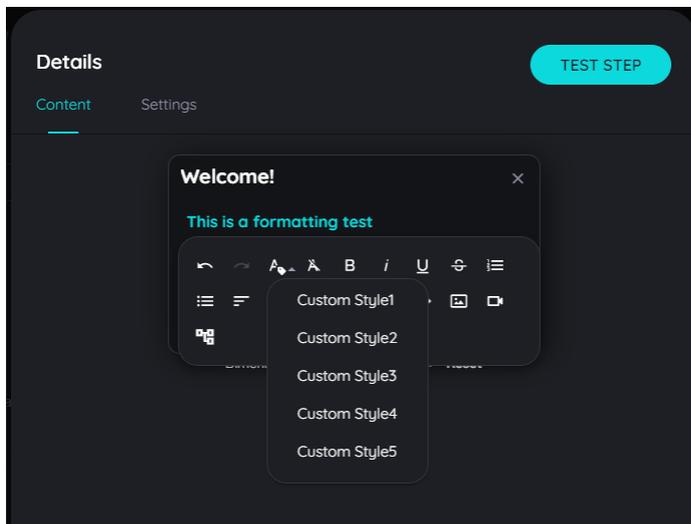
In einigen Organisationen ist es erforderlich, dass das CI der Organisation verwendet werden wird (bspw. die Schriftart Arial als Standardschriftart).

Um diese Anforderung zu unterstützen, bietet AppNavi die Möglichkeit, bis zu 5 benutzerdefinierte Stilvorlagen im Portal festzulegen. Diese können anschließend Planner für die Gestaltung von Routen- und Hotspot-Inhalten verwendet werden.

Um eine benutzerdefinierte Stilvorlagen zu erstellen, können die folgenden 5 Klassen im Abschnitt Custom CSS der Anwendung definiert werden:

- > .an-custom-style-1
- > .an-custom-style-2
- > .an-custom-style-3
- > .an-custom-style-4
- > .an-custom-style-5

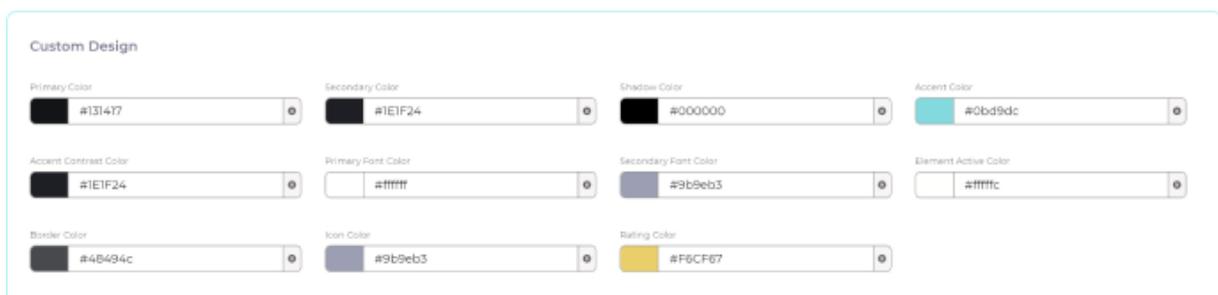
Jede der oben genannten Klassen legt fest, wie die entsprechenden Inhalte im Planner dargestellt werden.



10.2 Custom Theming

Im Profil einer App können bei aktiviertem Custom Theming in Summe 11 Farben für das Farbschema von AppNavi definiert werden. Darüber hinaus besteht die Möglichkeit, dass mit Hilfe von Custom CSS noch detailliertere Änderungen am UI vorgenommen werden.

Colorpicker





- > **Primäre Farbe**
Die Primärfarbe wirkt sich auf das grundlegende Aussehen des AppNavi-Menühintergrunds sowie auf den Hintergrund der Tooltips, Hotspots, Neuigkeiten und Ankündigungen aus.
- > **Sekundärfarbe**
Die Sekundärfarbe wirkt sich auf das grundlegende Erscheinungsbild der AppNavi-Routenkacheln aus.
- > **Schattenfarbe**
Die Schattenfarbe wirkt sich auf alle Schatten in AppNavi aus. Zum Beispiel die Schatten für die Routenkacheln.
- > **Akzentfarbe**
Die Akzentfarbe setzt Akzente in AppNavi. Sie beeinflusst die Farbe von Schaltflächen (z.B. in Tooltips, im Routenplaner oder im AppNavi-Menü) und anderen hervorgehobenen Elementen.
- > **Akzent-Kontrastfarbe**
Die Akzentkontrastfarbe hebt sich von der Akzentfarbe ab.
- > **Primäre Schriftfarbe**
Die primäre Schriftfarbe wird in der Regel in Bereichen mit der primären Farbe verwendet, z.B. auf den Routenkacheln, im AppNavi-Menü.
- > **Sekundäre Schriftfarbe**
Die sekundäre Schriftfarbe wird in der Regel in allen anderen Bereichen verwendet.
- > **Element-Aktiv-Farbe**
Die Element-Aktiv-Farbe wird verwendet, um aktive oder ausgewählte Elemente hervorzuheben.
- > **Farbe des Rahmens**
Die aktive Farbe des Rahmens wird für alle Rahmen verwendet.
- > **Icon-Farbe**
Die Icon-Farbe beeinflusst das Aussehen aller Icons in AppNavi, zum Beispiel die Icons auf den Routenkacheln.
- > **Farbe der Bewertung**
Die Bewertungsfarbe steuert die Bewertungssterne auf den Routenkacheln.

10.3 Custom Coding

Mit Hilfe eines SDKs stellt AppNavi die AppNavi Client API zur Verfügung. Dieses Set an Methoden und Events erlaubt es die Funktionalität von AppNavi mit der Zielapplikation zu verbinden und zu erweitern.

Innerhalb der AppNavi Architektur kann an zwei Stellen Custom Code zum Einsatz kommen. Zum einen im Profil einer Applikation, zum anderen in einzelnen Schritten innerhalb des Profils einer Route. Die folgenden Optionen stehen zur Verfügung:

| Allgemein

- > `setGuestUserId` :method
- > `setPreferredLanguage` :method
- > `hideAvatar` :method
- > `showAvatar` :method
- > `getCurrentApplicationId` :method
- > `getCurrentApplication` :method
- > `getCurrentSubscriptionId` :method
- > `getApplicationContainer` :method
- > `setTargetAudienceConditions` :method
- > `avatarLoad` :event

| Ankündigung

- `getAvailableNews` :method
- `showAnnouncement` :method
- `newsShow` :event
- `newsRead` :event
- `announcementShow` :event



| Hotspots

- > getAvailableHotspotCollections :method
- > getPageHotspots :method
- > hidePageHotspots :method
- > showPageHotspots :method
- > hideHotspotItem :method
- > showHotspotItem :method
- > startHotspotAction :method
- > stopHotspotAction :method
- > startRenderHotspots :method
- > stopRenderHotspots :method
- > hotspotActionStart :event
- > hotspotActionStop :event

| Automation

- > insertValue :method
- > skipAutomation :method
- > Analytics
- > setUserRole :method

| Route

- > startRoute :method
- > getActiveRouteId :method
- > getCurrentStepId :method
- > getStartMode :method
- > stopRoute :method
- > playStep :method
- > getCurrentElement :method
- > getAvailableRoutes :method
- > getTooltipContainer :method
- > routeStart :event
- > routeStop :event
- > playStep :event



11 Datenschutz

Grundsätzlich basiert die Funktionalität von AppNavi auf einem „datenblinden“ Konzept. D.h AppNavi Routen werden im Zielsystem gestartet und kennen dort den Kontext der zu findenden Elemente (Element-Typ, Attribute, Klassen, ungefähre Position, uvm.). Vollständig unbekannt ist jedoch der Inhalt der Elemente. So werden keine Input-Values, Texte, o.ä. gespeichert.

Zur Laufzeit ermittelt der Fuzzy-Score-Algorithmus das Element mit der höchsten Übereinstimmung von aufgezeichneter Routendefinition und aktueller Bildschirmsituation.

Die Funktionalität von AppNavi wird dabei (inkl. der Segmentierungslogik) immer auf Seite des Clients ausgeführt – ein Transfer von Informationen an das Backend von AppNavi findet zu keinen Zeitpunkt statt.

12 IT-Security

AppNavi wird als SaaS Lösung regelmäßig durch externe Pentests geprüft. Diese orientieren sich an gängigen Standards wie bspw. OWASP und prüfen Kriterien wie bspw. Session Management, File Upload und Download, Cross-Site Scripting, SQL Injection, Path Traversal, Parametermanipulation sowie weitere OWASP Top 10 Vulnerabilities.

Zusätzlich zu den von AppNavi selbst veranlassten Audits wurde AppNavi auch in zahlreichen weiteren Tests bei Kundeneinführungen geprüft.

| Berechtigungskonzept

- > Zugangsebene nach dem **Need-to-know-Prinzip**
- > Die Benutzerberechtigungen werden von den jeweiligen Abteilungen definiert und vergeben
- > Es stehen dedizierte Benutzerrollen zur Verfügung

| Datenschutz

- > Die Daten werden stets verschlüsselt - während der Speicherung ("at rest") sowie der Übertragung ("at transit")
- > Die Integrität der Informationen wird durch die Protokollierung und Versionskontrolle aller Benutzeränderungen gewährleistet
- > Die Daten werden in einem AWS-Rechenzentrum (Frankfurt) gespeichert

Es werden keine persönlichen Daten gespeichert. Die Analysedaten werden pro Benutzerpopulation ausgewertet. Es ist nicht möglich, Rückschlüsse auf einzelne Benutzer zu ziehen.

13 Software-Ergonomie

AppNavi und all seine Komponenten sind auf möglichst einfache und intuitive Benutzbarkeit ausgelegt.

Auf diesem Gebiet wurden verschiedene Prüfung bei Konzernbetriebsräten erfolgreich bestanden. Der Fokus hier liegt auf Themen der Softwareergonomie wie:

- > Bedienbarkeit
- > Konsistenz
- > Barrierefreiheit

Im Rahmen der Entwicklung folgen wir den Standards und Vorgaben gemäß DIN EN ISO 9241 (Teile 11, 13-16, 110, 112, 125, 129, 143, 151, 154, 161, 171, 303, 306) und WCAG 2.1 Level AA (ISO/IEC 40500:2012).



14 Grenzen und Limitierungen

Für die Nutzung von AppNavi gelten folgende Grenzen und Limitierungen. Diese können bei Bedarf angepasst werden:

Tenant	
Max. Workspaces pro Tenant	100
Max. Bilder pro Tenant	1GB
Max. Bildgröße	3MB
Max. Autoren pro Tenant	100
Max. Routen pro Tenant	250
Max. News pro Tenant	250
Max. Applikationen pro Tenant	100
Max. Route-Collections pro Tenant	50
Max. Länge Titel	50
Max. Länge Company	100
Max. Anzahl Backups	5GB
Max. Anzahl User Profiles	50.000

Applikation	
Max. Anzahl zugewiesener Routen	50
Max. Anzahl zugewiesener News	50
Max. Anzahl zugewiesener Route-Collections	25
Max. Länge Titel	50
Max. Länge Kategorie	50
Max. Länge Muster	128
Max. Länge Url	128
Max. Länge Custom Code	15.000
Max. Länge Custom CSS	15.000
Max. Größe Bild	3MB
Zulässige Bildformate: .jpg, .jpeg, .png	3
Max. Anzahl publizierter Routen	30
Max. Anzahl publizierter News	50
Max. Anzahl publizierter Route-Collections	10

News	
Max. Länge Titel	50
Max. Länge Beschreibung	100
Max. Länge Inhalt	1.000
Max. Größe Bild	3MB
Zulässige Bildformate: .jpg, .jpeg, .png	3

Benutzer	
Max. Länge Vorname	25
Max. Länge Nachname	25
Max. Länge Telefonnummer	25
Max. Länge Email Adresse	50
Max. Länge Benutzername	25

Workspaces	
Max. Routen pro Workspace	50
Max. Applikationen pro Workspace	50
Max. News pro Workspace	100
Max. Route-Collections pro Workspace	25
Max. Autoren pro Workspace	100
Max. Länge Titel	50
Max. Länge Beschreibung	100

Routen	
Max. Anzahl Schritte	30
Max. Länge Titel pro Schritt	100
Max. Länge Inhalt pro Schritt	1.000
Max. Größe pro Schritt	100KB
Max. Länge Custom Code (OnBeforeRender) / Schritt	5.000
Max. Länge Custom Code (OnAfterRender) / Schritt	5.000
Max. Anzahl Captures pro Schritt	3
Max. Länge Url (relativer Pfad)	512
Max. Größe Route	3MB
Max. Länge Beschreibung	100

Routen-Collections	
Max. Länge Titel	50
Max. Länge Beschreibung	100
Max. Länge Inhalt	1.000
Max. Größe Bild	3MB
Zulässige Bildformate: .jpg, .jpeg, .png	3
Max. Anzahl zugewiesener Routen	15

Daten-Vorhaltefristen (in Monaten)	
Logs (technical logs)	
UBM analytics data	
Content analytics data	
Tenant data (incl. all user-related and analytics data) after the end of the contract.	

Insights	
Anzahl der monatlichen Events für das User Behavior Mining (nach Erreichen der Grenze schaltet sich die Aufzeichnung ab und setzt sich im nächsten Monat fort)	100.000



15 Weitere Leistungsbereiche

Neben der Bereitstellung von Softwareprodukten bietet AppNavi weitere Service-Leistungen an:

| SW-Implementierung

- > **Beratung:** Anforderungsaufnahme, Adoption Konzeption, Learner Journeys, KPI Definition, etc.
- > **Set-Up Services:** Integration, Customizing, Styling, Segmentierungen, Inhaltserstellung
- > **Training:** Training der Autoren
- > **Go Live Support**

| SW-Pflege

- > **Standard-Produkt-Support:** Bereitstellung von Releases, Störungsbehebung (Incident-Management)
- > **Optionales Supportleistungen:** 1st/2nd Level für Standard-Applikationen, umfassendere Servicezeiten, individuelle Service-Level, weitere Supportsprachen, dedizierter Support