Whitepaper

Die nXt-Plattform & Technologie

Konzepte und Systemarchitektur

Stand: Februar 2023

www.aeb.com

Inhalt

1. Die nXt-Plattform

1.1 Zentrale Anforderung: Flexibilität

Logistische Prozesse sind eng gekoppelt mit den physischen und regionalen Gegebenheiten der Unternehmen, den Eigenschaften ihrer Produkte und den Prozessen ihrer Kunden und Lieferanten. Kein Wunder also, dass sich diese Prozesse in den Unternehmen zwar oft ähneln, aber nie 100 % gleich sind.

Sie mit einer reinen Standardsoftware zu unterstützen bzw. zu automatisieren, führt daher in der Regel zu suboptimalen Lösungen und Unzufriedenheit bei den Anwendern. Individuell für einzelne Unternehmen entwickelte Softwarelösungen sind dagegen passgenauer, aber unangemessen teuer.

AEB-Softwarelösungen zeichnen sich von jeher durch ihren Standardsoftware-Charakter bei gleichzeitiger kundenindividueller Anpassbarkeit aus. Dies gelingt durch vielfältige Möglichkeiten zur Konfiguration der AEB-Produkte, also durch Schalter, Einstellungen und Stammdaten. Darüber hinaus sind programmatische Erweiterungen möglich, also Programmierung von Zusatzfunktionen und Erweiterungen des Datenmodells. Und das, ohne die Releasefähigkeit/Updatefähigkeit der Standardsoftware zu behindern.

Also ist doch alles gut, oder?

Jein. Denn durch Globalisierung und Digitalisierung erhöht sich der Innovationsdruck auf Unternehmen seit Jahren immer weiter. Auch in den logistischen Prozessen ist eine permanente Optimierung notwendig, um sich im immer härter werdenden Wettbewerb zu behaupten und zu differenzieren. Hier sind in Programmcode gegossene Prozesse und statische Datenmodelle zunehmend zu unflexibel.

Zugleich macht es Sinn, auch in der Gestaltung von Benutzerschnittstellen (UI) von Unternehmenssoftware den Trend zu adaptieren, den wir alle aus dem privaten Umfeld schon seit Jahren kennen: Einfach zu nutzende und auf eine konkrete Aufgabenstellung reduzierte Anwendungen (Stichwort: Handy-Apps), die intuitiv verstanden und bedient werden können. Solche reduzierten Anwendungen erfordern kaum Schulung, Dokumentation oder Einarbeitungszeit und haben gerade in häufig vorkommenden Regelprozessen große Effizienzvorteile gegenüber traditionellen, mächtigen „Allzweck-Anwendungen“, die aufgrund ihrer Funktionsvielfalt nur für Power-User bzw. Experten und eher für Ausnahmebehandlungen und komplexe Bearbeitungen geeignet sind.

1.2 Die nXt-Technologie

Um auch unter diesen sich weiter verändernden Rahmenbedingungen weiterhin die besten Software-Lösungen für unsere Kunden zur Verfügung stellen zu können, haben wir bei AEB die nXt-Technologie entwickelt. In der AEB-Cloud werden damit Lösungen für Ihre Geschäftsprozesse erstellt, betrieben und iterativ-inkrementell („agil“) weiterentwickelt. nXt setzt neue Maßstäbe in der Flexibilität und Anpassbarkeit auf die individuellen Prozesse jedes einzelnen Kunden.

So können Prozesse, Apps, Datenmodell und APIs in weiten Teilen ohne Programmierung („Customizing“) konfiguriert und angepasst werden, und das im laufenden System, d. h. ohne Downtime.

Alle Konfigurationen einer Lösung werden versioniert verwaltet. Dadurch werden automatisch alle Änderungen der Lösung archiviert und dokumentiert und es kann jederzeit auf einen vorherigen, gesicherten Versionsstand zurückgerollt werden (z.B. falls durch eine nicht ausreichend getestete Anpassung ein Fehler im Produktivsystem auftritt).

„Klassische Softwareentwicklung“ findet nur bei Entwicklung neuer bzw. Änderung der Funktionalität bestehender Apps statt. Zahlreiche in die AEB-Cloud-Plattform integrierte Tools und Sammlungen wiederverwendbarer Komponenten unterstützen sowohl beim Erstellen der UI als auch bei der Implementierung von Geschäftslogik.

Grundlegende Konzepte

Process Driven Design: Der grafisch konfigurierte Geschäftsprozess, bestehend aus User-Tasks (Benutzerinteraktion) und Service-Tasks (automatisierte Hintergrundverarbeitung), steht im Mittelpunkt jeder nXt-Lösung.

UI/Benutzerdialoge: User-Tasks werden auf Basis von Persona-Analysen als User-Story mit Adobe XD® Prototypen konzipiert und als responsive HTML5-App implementiert. Sie unterstützen damit sowohl Desktop-PC als auch mobile Devices wie Scanner, Tablet oder Handy.

Datenmodell: Auf die Prozesse des Kunden optimiert und reduziert. Flexibel konfigurierbar, auch live im laufenden System. Dadurch hohe Flexibilität und Erweiterbarkeit, Vermeidung von Downtimes sowie optimierte Performance und verminderter Ressourcenverbrauch.

Business Services: Standardisierte (= für viele Kunden gleichermaßen nutzbare) Funktionen aus AEB-Produkten wie Zoll- und Carrieranbindungen können problemlos über APIs integriert werden.

Templates: Prozesse, Apps, und Datenmodelle müssen nicht für jeden Kunden neu entwickelt werden – umfangreiche Template-Sammlungen aus 40 Jahren AEB Erfahrung ermöglichen einen schnellen Start.

AEB Cloud only: Sicherer und performanter Betrieb in den Rechenzentren der AEB unter Einsatz modernster Technologien (Docker, Kubernetes).

Agiles Vorgehen: Optimiert für agile (iterativ-inkrementelle) Projektarbeit: Prozesse, Apps und Datenmodelle können auch im laufenden Produktivbetrieb jederzeit einfach weiterentwickelt, getestet und in den Produktivbetrieb genommen werden.

1.3 Standardsoftware vs. Individuelle Lösung

Die große Stärke der nXt-Plattform ist die Flexibilität, also die leichtgewichtige Anpassbarkeit der Lösung an die individuellen Prozesse. Dies umfasst auch ein auf die Aufgabenstellung optimiertes Datenmodell sowie die intuitive und effiziente Benutzerführung durch auf die jeweiligen Use-Cases zugeschnittene Apps. Führt das aber nicht zwangsläufig zu teuren Individuallösungen?

Ein doppeltes Nein. Warum?

Eine Template-Sammlung mit häufig vorkommenden bzw. idealisierten Prozessen, Apps und Datenmodellen ermöglicht, neue nXt-Lösungen zügig zu implementieren. Die Templates (Kopiervorlagen) werden dann in einem agilen Prozess (d. h. in kleinen, überschaubaren Schritten) an die speziellen Bedürfnisse des jeweiligen Kunden weiter angepasst. Dies senkt die initialen Projektaufwände deutlich und führt damit auch früh zu wertvollem Praxisfeedback, das dann in die weitere Optimierung der Lösung einfließen kann.

Darüber hinaus stehen in der AEB Cloud Business-Services zur Verfügung, die auf robusten, skalierenden und kostengünstigen Einsatz und Multi-Tenant-Betrieb optimiert sind. Diese Standardfunktionen werden über APIs

in übergreifende Geschäftsprozesse und insbesondere in nXt-Lösungen integriert. Beispiele für solche Business-Services sind die Zollanbindung (z. B. Export Filing: ATLAS), die Sanktionslistenprüfung (Compliance Screening) oder die Carrier-Anbindung (Carrier Connect) – und viele weitere mehr.

Daher muss in nXt-Lösungen z. B. eine Zollanbindung nicht für jeden Kunden erneut konzipiert, entwickelt und beim Zoll zertifiziert werden. Und auch die laufende Wartung und Weiterentwicklung dieser Business Services, z. B. aufgrund neuer Vorgaben des Zolls, erfolgt zentral durch AEB für alle AEB Kunden.

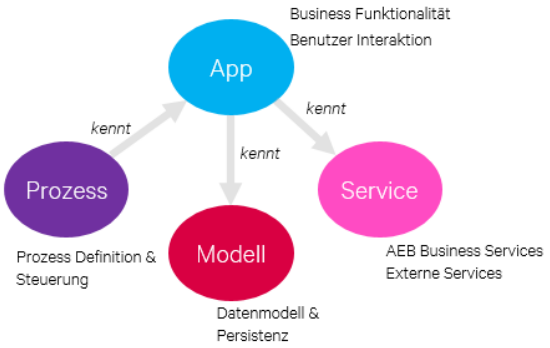
nXt-Lösungen in der AEB Cloud sind also weder eine reine Standardsoftware noch eine reine Individuallösung, sondern eine Mischung aus beidem. Vereinfacht gesagt: Individuell pro Kunde sind die Geschäftsprozesse mit den spezialisierten Apps, sowie das Datenmodell und die Integration in vor- und nachgelagerte Prozesse und IT-Systeme. Fachfunktionalitäten dagegen werden, wo immer möglich, durch Nutzung von AEB Business Services oder auch Services Dritter in die Lösung integriert.

Wir sagen dazu auch: „nXt-Lösungen sind kein Standard, aber basieren auf Standards“.

2. Die nXt-Technologie im Detail

Werfen wir nun einen detaillierteren Blick auf die Architektur und die wesentlichen Aspekte und Komponenten der nXt-Plattform und wie die nXt-Technologie ermöglicht, die oben genannten Herausforderungen optimal zu meistern.

2.1 Grundlegende nXt-Architektur



Während klassische Multi-Tier-Architekturen (Schicht-Architekturen) die Komplexität eines Softwaresystems verringern können, unterstützen Microservice-orientierte Konzepte eine fachliche Strukturierung sowie die iterative Entwicklung und damit die flexible Erweiterbarkeit des Gesamtsystems.

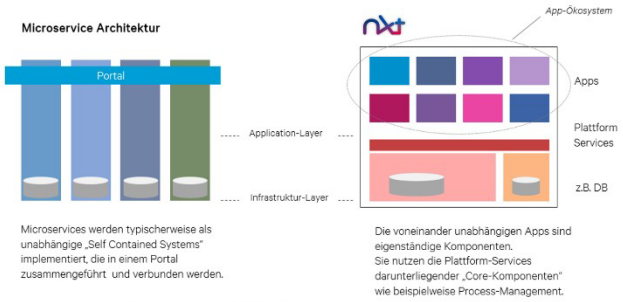
Die nXt-Architektur bedient sich aus beiden Grundkonzepten, um die Vorteile in dem vorgesehenen Lösungsumfeld zu vereinen. Die Hauptkomponenten einer nXt-Lösung, also die Apps als Träger der Anwendungslogik und der UI, die Geschäftsprozesse, das Datenmodell sowie die Business-Services sind voneinander entkoppelt und interagieren auf Basis stabiler Schnittstellen.

Apps bilden die modularen Bausteine einer nXt-Lösung im Sinne einer Microservice orientierten fachlichen Strukturierung einer Gesamtlösung. Gemeinsam nutzen alle Apps das Datenmodell, die Business Services sowie Funktionen der nXt-Plattform im Sinne einer Mehrschicht-Architektur.

Jede App innerhalb einer nXt-Lösung ist in dieser Analogie also ein eigener Microservice mit folgenden Microservice-typischen Eigenschaften:

- Die App kann einfach ersetzt werden
- Die App ist isoliert gegenüber anderen Apps
- Die App implementiert eine einzelne Geschäftsfunktion
- Die App ist für eine bestimmte Benutzergruppe (Persona) optimiert

Anders als in reinen Microservice-Architekturen, sind die Apps in nXt jedoch keine komplett eigenständigen „Self-contained Systems“, sondern setzen das Vorhandensein der nXt-Plattform voraus. Für die Datenhaltung bedeutet das beispielsweise, dass nicht jede App ihre eigene Datenbank mit sich bringt, sondern für das Speichern und Abrufen von Daten den dafür vorgesehenen nXt-Plattform-Service nutzt. nXt-Apps funktionieren daher nur eingebettet in eine nXt-Lösung.



Komponenten-Architektur

2.2 Process Driven Design

Dreh- und Angelpunkt jeder nXt-Lösung ist die Beschreibung der abzubildenden fachlichen Geschäftsprozesse in der BPMN-Notation. Diese Beschreibung wird schrittweise bis hin zu den einzelnen Arbeitsschritten und Benutzerrollen verfeinert.

BPMN (Business Process Model and Notation) ist eine standardisierte, ausführbare Sprache zur Beschreibung von Geschäftsprozessen und Arbeitsabläufen. Die Notation erfolgt visuell, d. h. durch Anordnung von grafischen Elementen.

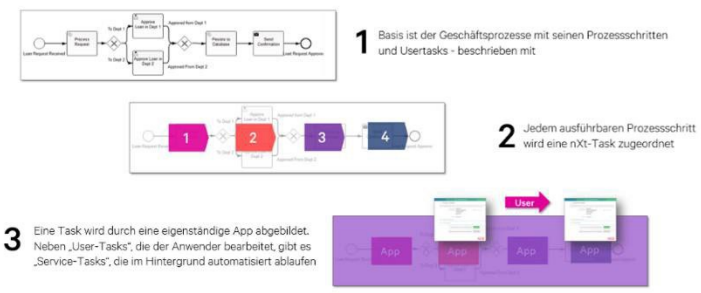
Die Notation ist in den letzten Jahren sehr populär geworden und steht bei vielen Hochschulen auf dem Lehrplan. BPMN gilt als wichtiges Element zur Vereinigung von IT- und Fachabteilung (Business IT Alignment).

BPMN-Definitionen basieren technisch auf XML. Die Ausführung erfolgt in Prozess- und Workflow-Engines. Herausgeber des Standards ist die Object Management Group (OMG), die auch die Unified Modeling Language (UML) verantwortet.

Als grafisches Modellierungswerkzeug wird hierzu der in nXt integrierte Prozesseditor verwendet. Diese fachliche Prozessbeschreibung stellt gleichzeitig die technische Prozessdefinition für die Steuerung der Abläufe innerhalb von nXt dar.

Ein Prozessschritt (Aufgabe) im Prozessmodell wird zu einem Task – ein Task wird mit genau einer App realisiert bzw. bearbeitet. Dieses simple Prinzip, durchgängig angewendet, macht eine nXt-Lösung einfach verständlich: Der Spezialist für Geschäftsprozesse findet für jeden Task eine entsprechende App, der Anwender kann seine Apps, seine Aufgaben direkt im Prozessschaubild zuordnen.

Es gibt zwei Arten von Tasks. User-Tasks sind interaktiv, d. h. haben eine UI und werden vom Anwender bedient. Service-Tasks sind für automatisierte Hintergrundverarbeitungen zuständig.



Process Driven Design

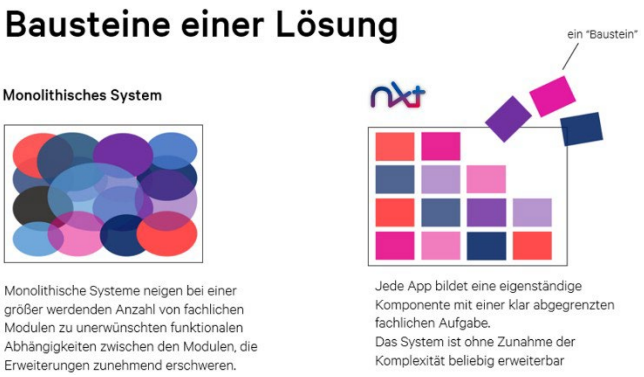
Gesteuert und orchestriert werden Tasks durch den zentralen Process Manager der nXt-Plattform, der auch die Rolle eines „Prozess-Überwachers“ innehat und sicherstellt, dass keine Aufgaben liegen bleiben.

2.3 Apps: Bausteine einer nXt-Lösung

Ein Task (Prozessschritt) eines nXt-Geschäftsprozesses wird durch genau eine App implementiert. Eine App übernimmt daher typischerweise genau eine fachliche Aufgabenstellung. Der Ansatz klein geschnittener Lösungsbausteine wurde als Konstruktionsprinzip bewusst gewählt, um auch bei wachsenden Anforderungen eine einfache und transparente Modularisierung des Systems zu erreichen und zugleich eine intuitive Bedienung der Lösung durch die Benutzer ohne aufwendige Schulungen zu ermöglichen.

Zur Laufzeit können neue Apps geladen werden, bestehende Apps durch eine neue Version aktualisiert oder aber aus einer Lösung entnommen werden. Damit sind Apps zentrale Bausteine für eine kontinuierliche und agile Weiterentwicklung einer nXt-Lösung.

Voraussetzung hierzu ist ein weiteres wichtiges Konstruktionsprinzip: Eine App soll eine möglichst autarke Funktionseinheit bilden und selbst wiederum nicht von anderen Apps abhängig sein.



Das schließt beispielsweise aus, dass eine App direkt Funktionen einer anderen App aufruft. Eine Kommunikation findet nur indirekt, entweder über das gemeinsam genutzte Datenmodell oder über Ereignisse in der Prozessverarbeitung statt. Damit kommt der nXt-Plattform eine zentrale Rolle zu: Sie bildet mit ihren Services die Laufzeitumgebung, in der Apps betrieben werden. Auch dadurch können Apps in der Regel sehr schlank gehalten werden, da man sich bei ihrer Implementierung auf die konkrete Businesslogik konzentrieren kann und die nXt-Plattform die typischen Standardaufgaben wie Persistenz oder Benutzersteuerung übernimmt.

Der architekturelle Ansatz von kleinen „Funktions-Bausteinen" harmoniert mit den Prinzipien moderner Anwendungsgestaltung, insbesondere auch einer aufgabenorientierten Benutzerinteraktion (1-1-3 Prinzip).

Das 1-1-3 Prinzip

Eine gute und intuitive Lösung gestalten heißt, eine Antwort darauf zu finden, WIE ein Anwender (1) eine Aufgabe (1) am einfachsten und schnellsten (max. 3 Klicks) lösen kann.

Dafür gilt es die 5 W's zu verstehen: WER soll die App nutzen (welches Know-how, Motivation, Sorgen besitzt der Anwender) und WARUM braucht sie/er die App (Rolle/Aufgabe). WANN und WO soll die Aufgabe gelöst werden (Prozess, Kontext, Umgebung etc.). WAS wird dafür benötigt (Voraussetzungen, Daten, Informationen, etc.)

Auf dieser Basis kann die Lösung optimal auf die entsprechenden Anwender zugeschnitten und gestaltet werden.

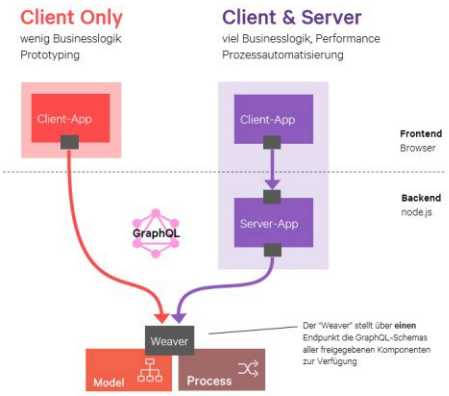
Eine App, bestehend aus Benutzeroberfläche und Business-Logik, wird zugeschnitten und optimiert auf die spezifische Aufgabenstellung des Benutzers. Varianten, beispielsweise für unterschiedliche Benutzergruppen, können entweder durch weitere spezifische Apps oder durch Varianten innerhalb einer App abgebildet werden.

Um auch in umfangreichen Lösungsumgebungen mit vielen unterschiedlichen Apps eine klare und intuitive Bedienung von nXt zu gewährleisten, wird die Auswahl der verfügbaren Apps pro Benutzer bzw.

Benutzergruppe gesteuert. Jeder Benutzer sieht nur die Apps, die er für seine Arbeit benötigt. Die Steuerung der Aufgaben- bzw. App-Bearbeitung übernimmt die zentrale Prozesssteuerung (Process Manager).

2.3.1 App Architektur

Apps bestehen aus einem Frontend- und einem Backend-Teil. Der Frontend-Teil läuft im Browser des Endgeräts des Anwenders und umfasst hauptsächlich das User-Interface, kann jedoch auch Teile der Anwendungslogik beinhalten. Der Backend-Teil läuft auf Servern in der AEB Cloud und enthält in der Regel den größten Teil der Fachfunktionalität der App. Beide Teile sind optional, das heißt es gibt sowohl Apps, die nur mit einem Frontend-Teil, als auch Apps, die nur mit einem Backend-Teil auskommen. Die oben beschriebenen Service Task Apps ohne UI haben beispielsweise keinen Frontend-Teil. Apps, die lediglich zur Anzeige von Daten genutzt werden und keine Daten verändern, benötigen hingegen meist keinen Backend-Teil.



Das Frontend einer App ist eine in TypeScript und Angular geschriebene Single Page Application (SPA). Das Backend einer App ist eine in TypeScript geschriebene node.js-Anwendung. Apps sind damit auch diejenigen Komponenten einer nXt-Lösung, in denen i. d. R. auch echte Softwareentwicklung in einer Programmiersprache notwendig ist.

Die nXt-Plattform enthält eine umfangreiche Sammlung von vorbereiteten UI-Komponenten, die eine effiziente Entwicklung von Apps ermöglicht. Die Verwendung dieser UI-Komponenten führt zu einer hohen Einheitlichkeit im Layout und den Bedienkonzepten der nXt-Apps und erleichtert damit die intuitive Bedienbarkeit durch die Benutzer. Die App-Konzeption wird jedoch nicht auf die Verwendung dieser vordefinierten UI-Komponenten eingeengt – es stehen daneben alle HTML5-Controls mit allen gestalterischen Freiheitsgraden zur Verfügung, um für spezielle Use Cases auch spezielle optimierte Bedienkonzepte umzusetzen.

TypeScript ist eine typisierte Erweiterung von JavaScript und ermöglicht damit eine mit anderen modernen Programmiersprachen vergleichbare Entwicklungsgeschwindigkeit und Code-Qualität.

Angular ist ein Open-Source UI-Framework zur Entwicklung von HTML5-Webanwendungen in TypeScript. Angular wird primär von Google entwickelt und legt großen Wert auf Qualität und Wiederverwendbarkeit von UI-Komponenten. Es hat sich in den letzten Jahren zu einem der wichtigsten UI-Frameworks auch im Bereich großer Enterprise-Anwendungen entwickelt. Durch Prinzipien wie Dependency Injection und ein ausgereiftes Tooling ermöglicht es effiziente und wartbare UI-Entwicklung.

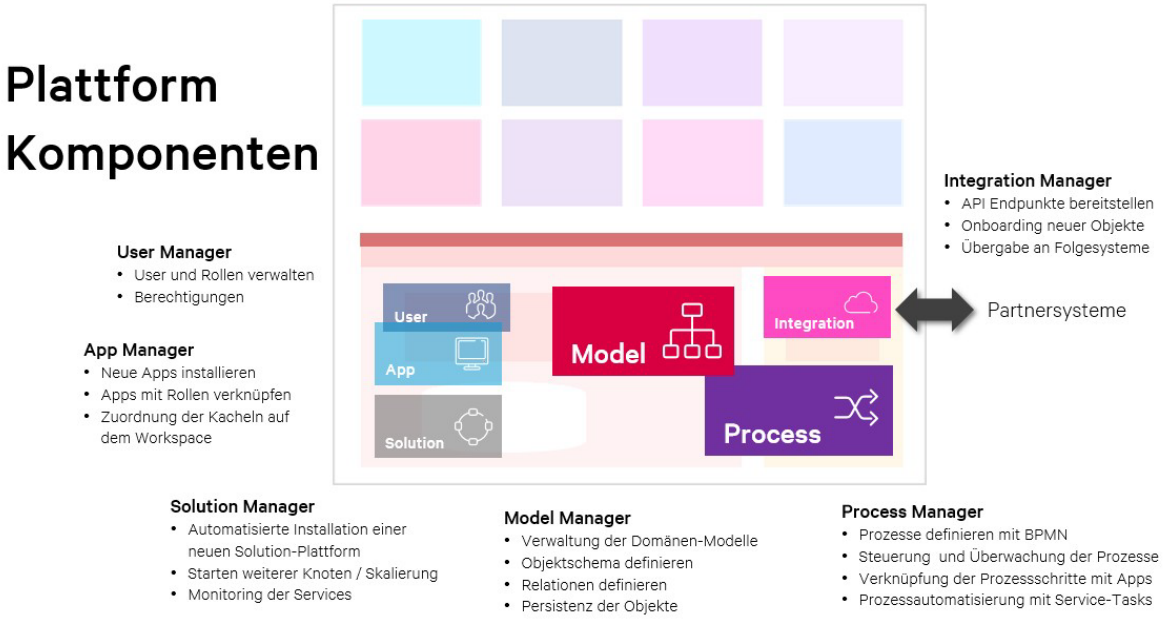
Eine Angular-Anwendung ist eine sogenannte Single Page Application (SPA). Das bedeutet, dass sie anders als bei klassischen Webanwendungen nicht aus vielen miteinander verlinkten HTML-Dokumenten besteht, sondern beim Aufruf der Anwendung im Browser nur eine einzelne Seite geladen wird, deren Inhalt sich dann jedoch während der Nutzung dynamisch ändert.

Node.js ist eine plattformübergreifende Open-Source-JavaScript-Laufzeitumgebung. Sie erlaubt es, den aus TypeScript automatisch erzeugten JavaScript-Code nicht nur im Browser, sondern auch serverseitig im Backend auszuführen. Dies hat den Vorteil, dass sowohl die Frontend- als auch Backenend-Entwicklung in derselben Programmiersprache (TypeScript) stattfinden kann. Node.js wurde mit besonderem Fokus auf Performance entwickelt. Anders als bei herkömmlichen Web-Serving-Techniken basiert Node.js auf dem Prinzip des Non-Blocking-I/O. Das bedeutet unter anderem, dass nicht für jede Anfrage ein eigener Thread benötigt wird, was wiederum zu einem geringeren Ressourcenverbrauch und schnelleren Antwortzeiten führt.

Des Weiteren gibt es für Node.js eine sehr große Zahl an Open-Source-Paketen, die über den Paketmanager NPM eingebunden werden können.

2.4 nXt-Plattformkomponenten

Wie oben dargestellt, funktionieren nXt-Apps nur eingebettet in eine nXt-Lösung, da sie auf grundlegende Funktionen der nXt-Plattform angewiesen sind. Die nXt-Plattform ist somit die Basis einer jeden nXt-Lösung. Auch die nXt-Plattform ist kein singuläres monolithisches System, sondern setzt sich aus einzelnen Plattformkomponenten zusammen, die jeweils wieder den Charakter von Microservices (analog zu den Apps) aufweisen. Im Folgenden werden die Plattformkomponenten und ihre jeweilige Aufgabe innerhalb einer nXt-Lösung näher beschrieben.



2.4.1 Kommunikation

Die Kommunikation der unterschiedlichen Komponenten einer nXt-Lösung erfolgt mittels GraphQL. Das heißt, sowohl die Kommunikation zwischen Front- und Backend einer App als auch die Kommunikation von Front- und Backend mit den darunterliegenden Komponenten der nXt-Plattform geschieht über GraphQL.

Ein wichtiges Prinzip von GraphQL ist, dass der Client dem Server in jeder Query exakt mitteilt, welche Daten er abrufen möchte. Anders als bei REST werden daher nur die benötigten Informationen (Felder) eines Objekts (Ressource) übertragen. Zudem kann im Vergleich zu REST in einer einzigen Anfrage ein ganzes Netz an Objekten abgerufen werden. In einer Query navigiert man sozusagen durch den Graph der Objekte und definiert genau, welche Objekte und welche Felder man abrufen möchte.

Dieses Vorgehen hat einige bedeutende Vorteile: Zum einen wirkt es sich positiv auf das Laufzeitverhalten der Anwendungen und die benötigte Netzwerk-Bandbreite aus. Ebenso wichtig sind jedoch die Vorteile dieses Vorgehens in Bezug auf die Versionierung von Schnittstellen. Da der Client definiert, welche Daten er benötigt, kann eine GraphQL-API auf Serverseite jederzeit erweitert werden, ohne dass bestehende Clients dabei berücksichtigt werden müssen.

GraphQL ist eine Open-Source Datenabfrage- und Manipulationssprache, die sich mehr und mehr zu einem verbreiteten Standard im modernen Web entwickelt und eine Alternative zu klassischen REST- oder SOAP-Schnittstellen ist.

Die Daten werden, wie auch bei REST üblich, im JSON-Format übertragen. Anders als bei REST ist eine GraphQL-Schnittstelle hingegen typisiert. D. h. jede GraphQL-Schnittstelle veröffentlicht, welche Objekte mit welchen Daten (Feldern) abgerufen werden können. Basierend darauf kann der Client dem Server in einer GraphQL-Query mitteilen, welche Daten er abrufen oder/und manipulieren möchte.

2.4.2 Model Manager

Jede nXt-Lösung besitzt ein individuelles Datenmodell. Dieses besteht weder aus vordefinierten Standardobjekten noch aus generischen Datenstrukturen. Die Objekte und Relationen des Datenmodells sind frei definierbar und können somit auf die jeweilige Aufgabenstellung und die damit verbundene Fachdomäne zugeschnitten und optimiert werden.

Wird nXt beispielsweise als Versandsystem eingesetzt, so können Objekte wie Delivery oder HandlingUnit definiert werden. Dabei wird festgelegt, welche Eigenschaften (Felder) ein Objekt hat und wie dieses in Relation zu anderen Objekten stehen kann. Um die Modellierung der Objekte zu erleichtern, existieren Vorlagen (Templates), aus denen man sich ganze Datenmodelle, einzelne Objekte oder auch nur Teile kopieren und dann weiter anpassen kann.

Die Definition des Datenmodells einer nXt-Lösung erfolgt mithilfe des Model Manager, der selbst Teil der Lösung ist und eine einfach zu bedienende UI besitzt, um Objekte, Felder, Relationen usw. definieren zu können. Des Weiteren ist der Model Manager für den „Betrieb“ des definierten Modells zuständig. Hierzu veröffentlicht der Model Manager eine typisierte GraphQL-API, die den lesenden und schreibenden Zugriff auf die modellierten Datenobjekte ermöglicht. Der Model Manager ist somit die Persistenzschicht einer nXt-Lösung und stellt die Basisfunktionen bereit, mit deren Hilfe die Apps Daten lesen und speichern können.

Unter der Haube speichert der Model Manager alle Daten in einer NoSQL-Datenbank (ArangoDB). Der Einsatz einer schemalosen Dokumentendatenbank hat gegenüber einer klassischen relationalen Datenbank mehrere Vorteile. Der wichtigste Vorteil ist die dadurch gewonnene Flexibilität. So lassen sich auch Änderungen am Datenmodell in einer nXt-Lösung ohne Downtime im laufenden Betrieb konfigurieren, ohne dass dabei, wie in einer SQL-Datenbank, das DB-Schema angepasst werden muss. Zudem ermöglicht es durch die Dokumentenstruktur eine deutlich intuitivere Abbildung der zu modellierenden Geschäftsobjekte.

ArangoDB (arangodb.com) ist eine Open Source Multi-Modell-Datenbank.

Unterstützt werden drei Datenmodelle: Dokumente, Graphen und Key-Value- Paare.

Daten werden in ArangoDB nicht in Form von relationalen Tabellen mit fest definierten Spalten abgelegt, sondern als JSON-Dokumente in sogenannten Collections gespeichert. Die Dokumente innerhalb einer Collection müssen dabei nicht zwangsläufig identisch strukturiert sein und enthalten jeweils auch nur die tatsächlich genutzten Daten und keine Platzhalter für leere Felder.

In nXt werden alle Daten eines zusammengehörigen Geschäftsobjekts innerhalb eines Dokuments in ArangoDB abgelegt. Da sich JSON-Dokumente beliebig tief schachteln lassen, ist dies intuitiv und problemlos auch dann möglich, wenn das Geschäftsobjekt eine Liste von Unterobjekten umfasst, wie es beispielsweise bei einer Lieferung mit Lieferungspositionen der Fall ist.

Um Dokumente zueinander in Beziehung zu setzen, können diese in ArangoDB wie in einem Graphen über Kanten miteinander verbunden werden. Jede Kante drückt dabei eine konkrete Beziehung aus, wie beispielsweise „(Lieferung) ist disponiert in (Sendung)".

ArangoDB besitzt eine eigene Abfragesprache namens AQL, mit der die Daten sowohl gelesen als auch geschrieben werden. AQL ist so aufgebaut, dass mit einer Query alle benötigten Daten abgerufen werden können, auch wenn diese über viele Collections verteilt sind. Dadurch kann der Model Manager einer nXt-Lösung jede GraphQL-Query in genau eine AQL-Query übersetzen, wodurch im Vergleich zum üblichen Aufteilen in viele einzelne Queries eine deutlich höhere Performance erreicht wird.

2.4.3 Process Manager

nXt ist ein prozessgesteuertes System. Das heißt, die Abläufe innerhalb des Systems folgen einem Prozessmodell. Dieses Prozessmodell steckt nicht implizit im Programmcode des Systems, sondern wird grafisch modelliert und anschließend ausgeführt. Es legt fest, welche Aufgaben (Prozessschritte) in welcher Reihenfolge im Leben eines Geschäftsobjekts durchgeführt werden müssen. Dem nXt-Prinzip zufolge werden diese Aufgaben dann mithilfe der Apps bearbeitet.

Die Modellierung der Prozesse erfolgt mit BPMN. Der Process Manager stellt hierfür einen entsprechenden grafischen Editor zur Verfügung. Darüber hinaus ist er verantwortlich für die Ausführung der modellierten Prozesse. Er weiß, welches Geschäftsobjekt sich in welchem Prozessschritt befindet und damit welche App momentan und als nächstes für die weitere Bearbeitung des Geschäftsobjekts zuständig ist. Des Weiteren können die laufenden Prozesse mithilfe des Process Managers überwacht und beeinflusst werden.

Der nXt-Process-Manager nutzt als Workflow Engine das bekannte und etablierte Camunda® BPM Framework.

2.4.4 Integration Manager

Wie beschrieben spielt die Modellierung der Geschäftsprozesse in einer nXt-Lösung eine wichtige Rolle. Häufig starten und enden die Geschäftsprozesse aber nicht in der nXt-Lösung, sondern in einem Vorsystem (z. B. ERP). Die einfache und reibungslose Prozessintegration und die Übergabe von Geschäftsobjekten von einem Vorsystem an nXt und zurück ist daher essenziell. Wird nXt beispielsweise als Versandsystem eingesetzt, entstehen die Lieferungen in einem vorgelagerten ERP-System und werden zu einem bestimmten Zeitpunkt zur

Versandabwicklung an die nXt-Lösung übergeben. Ist die Versandabwicklung abgeschlossen, findet eine Rückmeldung (z. B. für die Warenausgangsbuchung) an das Vorsystem statt.

Schnittstellen zu Partnersystemen werden in nXt mithilfe des Integration Managers eingerichtet. Er ermöglicht es, sowohl eingehende als auch ausgehende Kommunikationskanäle zu konfigurieren. Für die eingehende Kommunikation werden sogenannte Endpunkte angelegt. Ein Endpunkt definiert unter anderem das Protokoll und Format für den Datenaustausch (SOAP, REST, XML, JSON, GraphQL, …) und legt Verarbeitungs- sowie Validierungsregeln fest. Die so konfigurierten Endpunkte können dann von Partnersystemen aufgerufen werden, um Daten an die nXt-Lösung zu übergeben. Für die umgekehrte Richtung, wenn also nXt eine Verbindung zu einem Partnersystem aufbaut, werden im Integration Manager zusätzlich noch die erforderlichen Zugangsdaten hinterlegt.

2.4.5 App Manager

Wie der Name bereits vermuten lässt, ist der App-Manager zuständig für die Verwaltung der Apps innerhalb einer nXt-Lösung. Mit ihm lassen sich neue Apps installieren, konfigurieren und vorhandene Apps aktualisieren. Auch die Zugriffsberechtigungen, mit denen gesteuert wird, welche Benutzer auf welche Apps zugreifen dürfen, werden hier hinterlegt.

2.4.6 Solution Manager

Mit Hilfe des Solution Managers lässt sich eine nXt-Lösung („Solution“) administrativ verwalten. Eine wichtige Rolle spielt dabei das sogenannte Solution Repo. Hierbei handelt es sich um ein Git Repository, in dem die gesamte Konfiguration einer nXt-Lösung versioniert abgelegt ist (Datenmodell, Prozessmodell, Endpunkte, installierte Apps, …). Der Solution Manager ermöglicht es, diese Konfiguration zwischen den unterschiedlichen Installationen einer nXt-Lösung zu transferieren. Änderungen an der Konfiguration im Entwicklungs-/Testsystem werden somit auf schnelle und einfache Weise ins Produktivsystem übernommen.

Zudem stehen im Solution Manager weitere administrative Funktionen zur Verfügung, um Backups der nXt-Lösung anzulegen und wiederherzustellen und um Plattform-Updates einzuspielen.

2.4.7 User Manager

Aufgabe des User Managers ist die Verwaltung der Benutzer, die Zugriff auf eine nXt-Lösung haben. Benutzer können manuell angelegt werden oder mittels Identity Federation aus anderen Systemen übernommen werden. Benutzern werden Rollen zugewiesen, um ihnen Berechtigungen auf bestimmte Funktionalitäten zu geben.

2.5 Betrieb in der Cloud: Docker und Kubernetes

Anwendungen in der AEB Cloud müssen hohe Anforderungen an Ausfallsicherheit und Performance erfüllen. Das System muss auf Lastschwankungen und Hardwareausfälle überwacht werden, um ggf. durch Hochfahren zusätzlicher Knoten (Anwendungsserver) die Systemleistung aufrechtzuerhalten. Und dies im 24/7-Betrieb.

Mit klassischen Methoden der manuellen Installation von Softwarekomponenten durch Operations-Mitarbeiter ist in einem Hochleistungsrechenzentrum diese Betriebsqualität nur mit hohem Aufwand zu

gewährleisten. AEB setzt daher in seinen Rechenzentren auf Containerisierung der Softwarekomponenten durch Docker und eine automatisierte Verwaltung und Skalierung der Anwendungssysteme durch Kubernetes.

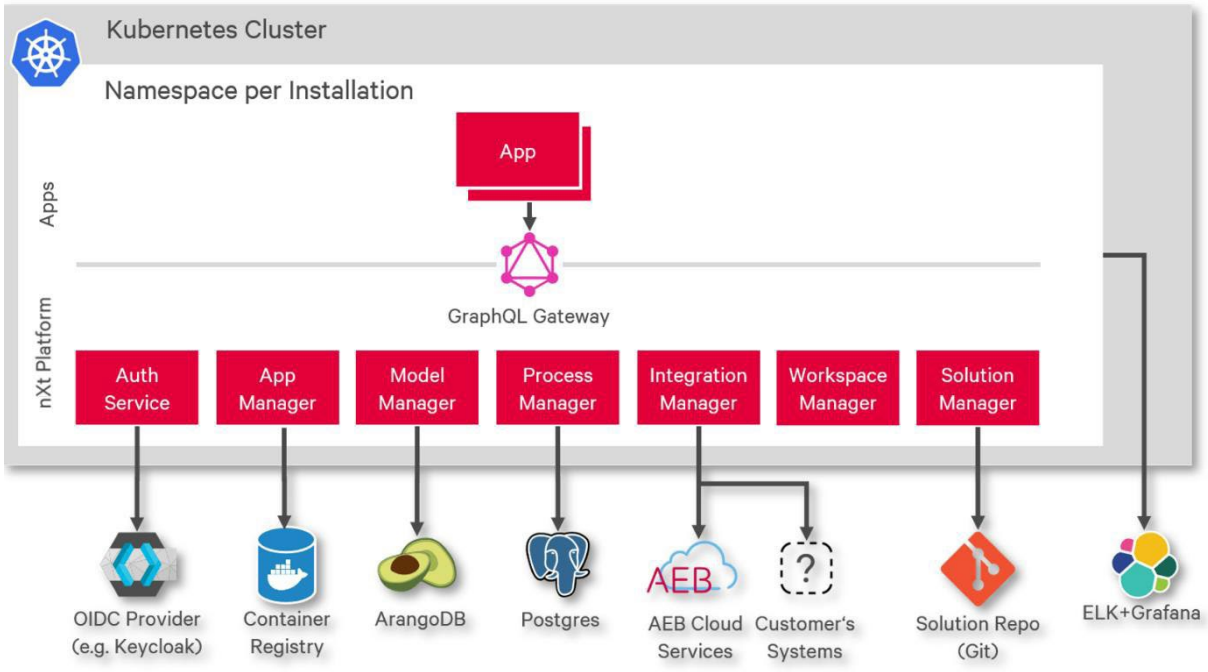
Docker vereinfacht das Deployment und den Betrieb komponentenbasierter Anwendungen und Dienste. Die einzelnen Komponenten eines Systems werden mithilfe von Docker in sogenannten Containern betrieben. Ein Docker-Container ist vergleichbar mit einer virtuellen Maschine, in der der zu betreibende Dienst installiert ist und läuft. Der Vorteil dieses Vorgehens besteht darin, dass der Dienst nicht mehr klassisch auf einem Server installiert und konfiguriert werden muss, sondern als vorkonfigurierter Container mitsamt seiner Laufzeitumgebung gestartet wird. Im Unterschied zu einer virtuellen Maschine auf Betriebssystemebene ist ein Docker-Container deutlich kleiner und ressourcenschonender.

Kubernetes ist eine Plattform zur Verwaltung von containerisierten Systemen. Mithilfe von Kubernetes lassen sich Docker-Container zu einem Gesamtsystem orchestrieren. Dabei können die Container auf mehrere Hosts verteilt, überwacht und automatisch skaliert werden. Die Orchestrierung mittels Kubernetes wird von führenden Cloud-Plattformen wie Microsoft Azure, Amazon AWS und vielen weiteren unterstützt.

Durch den Einsatz dieser Technologien ergeben sich unter anderem folgende Vorteile:

- Schnelle Softwareauslieferzyklen (Continuous Deployment)
- Einfache Skalierbarkeit, d.h. robuster und performanter 24/7-Betrieb auch bei starken Lastschwankungen und Hardware-Ausfällen

Das folgende Bild zeigt schematisch den Aufbau einer nXt-Lösung aus Betriebssicht:



Jede nXt-Kundenlösung erhält innerhalb von Kubernetes einen eigenen Namespace und ist damit auch auf der Betriebsebene getrennt von Lösungen anderer Kunden.

Innerhalb des Namespace befinden sich die Apps sowie die weiter oben beschriebenen Plattformkomponenten. nXt ist somit ein Single-Tenant-System, in dem selbst die nXt-Plattformkomponenten nicht über mehrere Kundenlösungen hinweg geteilt werden. Diese strikte Entkopplung sorgt für hohe Sicherheit in Bezug auf Datensicherheit und -schutz und maximale Robustheit und Performance im Betrieb. Zudem erlaubt dies, die nXt-Lösung auf die jeweiligen Bedürfnisse des Kunden optimieren zu können, sei es unter Betriebsaspekten oder auch in der fachlichen Implementierung der Lösung.

Jede einzelne App wie auch die Plattformkomponenten werden als separate Container betrieben und können somit unabhängig voneinander skaliert, versioniert und gewartet werden. Das GraphQL-Gateway ermöglicht es den Apps, auf alle Funktionen der nXt-Plattform über ein gemeinsames GraphQL-Schema zugreifen zu können. Die einzelnen Schnittstellen der Plattformkomponenten verschmelzen somit gegenüber den Apps zu einer gemeinsamen Solution API.

Außerhalb des Kubernetes-Namespace befinden sich darüber hinaus noch folgende Back-End-Systeme:

- OpenID Connect Provider (Keycloak): Ermöglicht Single-Sign-On (Identity federation)
- Container Registry: Hier liegen u. a. die vorbereiteten Docker Container (genauer: Images) der Apps
- ArangoDB: Der Datenbankserver, über den der Model Manager die Daten einer nXt-Solution speichert
- Postgres: Ein klassischer SQL-Datenbankserver, mithilfe dessen der Process Manager seine Daten speichert
- AEB Cloud Services: Die weiter oben beschriebenen Business Services der AEB wie beispielsweise Zollanbindung, Sanktionslistenprüfung, Carrier-Anbindung und viele mehr. Auch non-AEB-Services können über deren API integriert werden
- Solution Repo: Ein Git-Repository, in dem alle Einstellungen der nXt-Solution versioniert abgelegt werden. Ermöglicht zudem die Auslieferung von Änderungen vom Test- ins Produktivsystem
- ELK+Grafana: Ein System, in dem alle Logging- und Monitoring-Daten einer nXt-Solution abgelegt und Betriebszustände und -trends visualisiert werden

2.6 Sicherheit, Authentifizierung, Single-Sign-on

Alle Plattformkomponenten und Apps werden von einer einheitlichen Authentifizierungsschicht geschützt. Über ein Rollenkonzept wird festgelegt, auf welche Daten, Apps und Funktionen die Benutzer zugreifen können. Die Anmeldung der Benutzer erfolgt im zentralen Identity- and Access-Management-System der AEB, das via OpenID-Connect an die nXt-Lösung angebunden ist. Die nXt-Lösung kann dabei mit anderen Produkten gekoppelt werden, sodass sich ein Nutzer nur einmal anmelden muss, um auf alle AEB-Anwendungen zugreifen zu können. Darüber hinaus kann Identity Federation via OpenID-Connect eingerichtet werden, um ein nahtloses Single-Sign-On zu ermöglichen und die Nutzer und Rechte beispielsweise aus dem Active Directory des Kunden zu übernehmen.

2.7 Open Source

Open Source Software findet sich heute in nahezu allen Softwarelösungen. Die Motivation für den Einsatz solcher Komponenten ist dabei weniger, dass sie kostenlos genutzt werden können, sondern vielmehr, dass Open Source Software durch den offengelegten Quellcode und die kollaborative Weiterentwicklung in

entsprechenden Communities rasch funktional leistungsfähige und qualitativ hochwertige Komponenten hervorbringt und diese leicht auffindbar und einsetzbar sind. Damit verbreiten sich viele Open Source Komponenten rasch und führen damit auch zu einer Verbreitung entsprechender Entwickler-Skills und Tutorials.

Auch die nXt-Plattform nutzt diverse Open Source Komponenten wie z. B. das weit verbreitete UI-Framework Angular und profitiert damit direkt von deren Qualität und den permanenten funktionalen Weiterentwicklungen.

AEB stellt zudem Teile der nXt-Plattform im Web als Open Source Komponenten der Allgemeinheit auf GITHUB (https://github.com/AEB-labs) zur Verfügung und lädt damit Softwareentwickler weltweit dazu ein, diese Komponenten einzusetzen und sie durch Feedback und konkrete Mitwirkung weiter zu perfektionieren.

3. Ausgewählte Business Cases

Die vorgestellten Konzepte machen die AEB Cloud mit der nXt-Plattform zu einem mächtigen und zugleich einfach nutzbaren Werkzeug, um Geschäftsprozesse in all ihrer Individualität passgenau und effizient in benutzerfreundlichen, performanten und sicheren IT-Lösungen abzubilden und bei Bedarf leichtgewichtig an geänderte Anforderungen anzupassen.

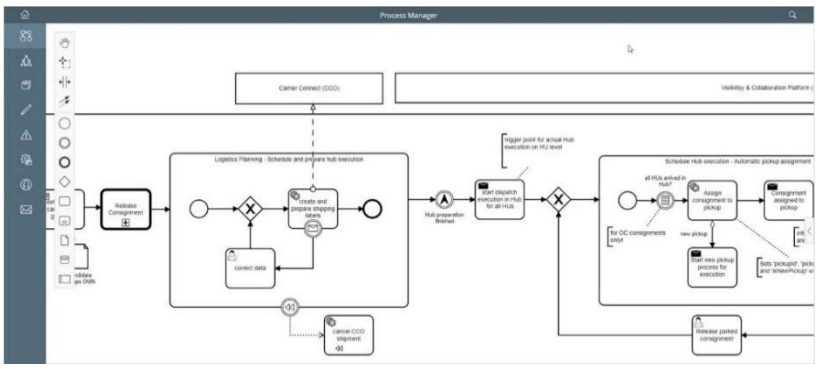
Anhand einiger konkreter Lösungsszenarien aus der Praxis soll nun der Transfer dieser Konzepte auf reale Aufgabenstellungen in typischen logistischen Geschäftsprozessen veranschaulicht werden. Es handelt sich hierbei um anonymisierte und teilweise vereinfachte Darstellungen, die aber realen Kundenlösungen entsprechen, die auf der AEB-Cloud-Plattform entwickelt wurden und sich bei AEB-Kunden im produktiven Echteinsatz befinden.

3.1 Szenario 1: Mehrstufige globale Distributionslogistik

Wir betrachten einen Hersteller eines breiten Spektrums an Messgerätetechnik, welche weltweit Anwendung in Industrieanlagen verschiedener Branchen wie Chemie, Energie und Kraftwerke, Lebensmittel, Wasser usw. findet. Produziert werden die Geräte an verschiedenen Standorten, vornehmlich in Europa.

Die Produkte werden über ein Logistiknetz, bestehend aus Verteilzentren (sogenannten Hubs), an Kunden auf der ganzen Welt geliefert.

Aufgrund der hohen Sendungsvolumina und zeitkritischer Lieferabrufe sollen über hochautomatisierte Prozesse Liefertermine im Voraus berechnet, dem Kunden genannt und am Ende auch eingehalten werden.



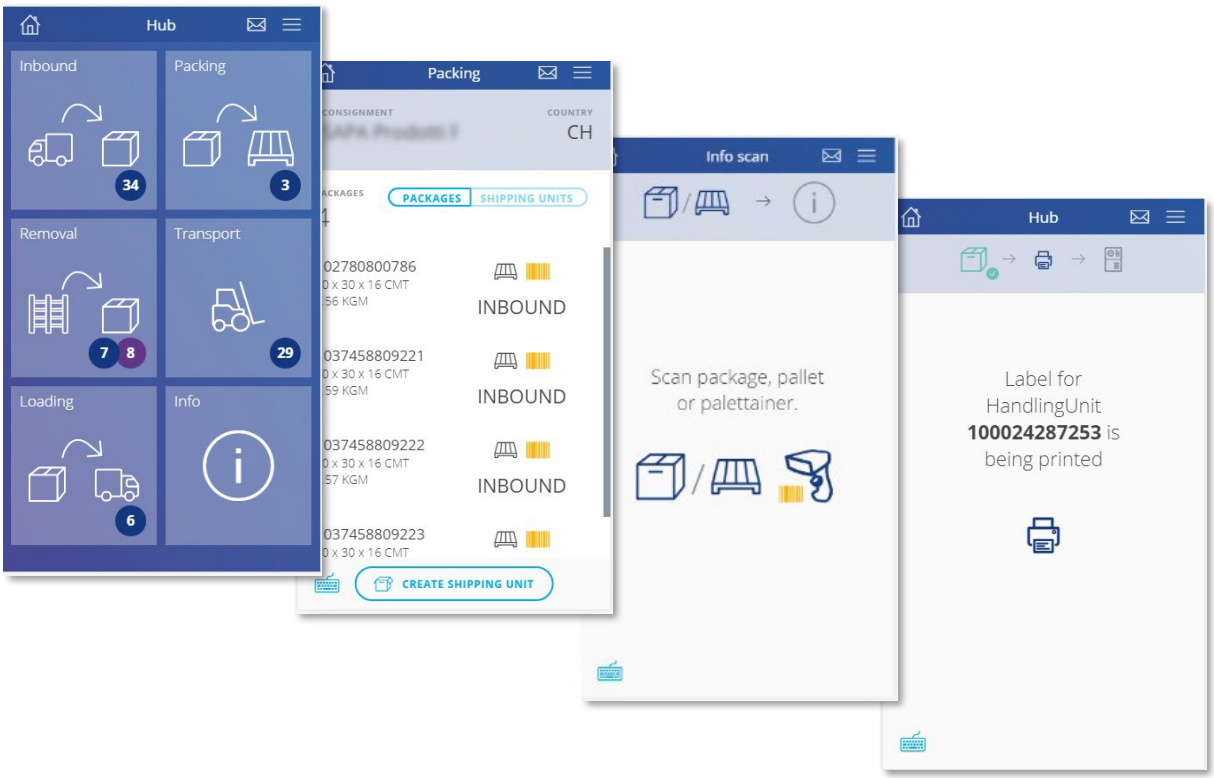
Warum ist eine nXt-Lösung für diese Aufgabenstellung besonders gut geeignet?

Der abzubildende Prozess umfasst u. a. das Initialisieren von Lieferungsdaten aus Auftrags- und Stammdaten, das Auswerten von Regelwerken (Business Rules) sowie die Integration zahlreicher Business Services sowohl von AEB als auch von Dritten.

Eine hohe Kundenorientierung in Verbindung mit der Forderung nach Kosteneffizienz verlangt ständige Optimierung und Veränderung der Prozesse. Die hochautomatisierten Abläufe müssen daher flexibel angepasst und auf neue Länder und Regionen ausgerollt werden können. Die im nXt-Prozess-Manager erstellten Prozessdefinitionen geben Auskunft über den aktuellen Stand sowie historische Versionen. Ständige Verbesserung wird gelebt.

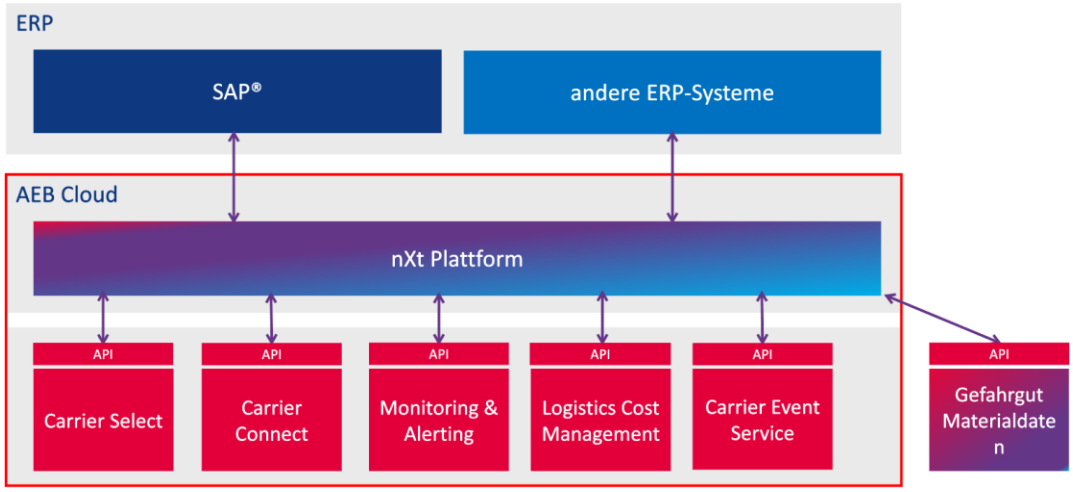
Der Einsatz von moderner Hardware im Logistikbereich ermöglicht dem Unternehmen ein sogenanntes „Hub out of the Box"-Szenario. Das bedeutet, dass Standorte für die Verteilung von Waren schnell und unkompliziert (über Bereitstellung der erforderlichen Hardware in einer Box bzw. einem Koffer) einbezogen werden können. Inhalt des Koffers: Ein mobiles Datenerfassungsgerät (Android Handheld) sowie ein tragbarer Label-Drucker. Somit braucht es nur noch einen funktionalen Raum sowie eine mobile Datenverbindung bzw. WLAN.

Die Benutzerdialoge (Apps) sind einfach und intuitiv gehalten, sodass keine lange Einarbeitung, Schulung oder Dokumentation erforderlich ist. Das ist gerade auch bei einem Einsatz auf verschiedenen Kontinenten und durch Benutzer mit unterschiedlichen Wissensständen und Kulturen besonders wichtig.



Welche AEB Business Services kommen zum Einsatz?

- Ermittlung von Routen, Laufzeiten, Versandarten, Spediteuren, Services etc. -> Carrier Select
- Aufbereitung von TDL-Labels und EDI-Nachrichten -> Carrier Connect
- Transparenz und Nachverfolgung der Auftragserfüllung und der damit verbundenen Transport- und Lieferketten -> Monitoring & Alerting
- Ermittlung, Kontrolle und Weiterberechnung von angefallenen logistischen Kosten (Transport, Lagerung, Handling etc.) -> Logistics Cost Management
- Verwaltung von Event-Nachrichten im Logistikprozess -> Carrier Event Service
- Ermittlung von Materialeigenschaften (zum Beispiel für Gefahrgut) -> Kundeneigener Materialservice



3.2 Szenario 2: Prüfung von eCommerce-Paketsendungen gegen Sanktionslisten

Hier betrachten wir einen führenden Anbieter von weltweiten Logistikleistungen und Services im B2B- und insbesondere auch im B2C-Umfeld mit entsprechend hohen Sendungsvolumina. Durch den Trend zum eCommerce (Webshops usw.) ist für die nächsten Jahre mit weiter deutlich ansteigenden Mengengerüsten (Anzahl Pakete) zu rechnen.

Bei Beförderung der Waren ist internationales Handelsrecht zu beachten. Aus diesem Grund werden die Adressdaten einer Sendung (Warenempfänger, Rechnungsempfänger, …) gegen geltende personen- und unternehmensbezogene Sanktionslisten und ggf. länderspezifische Embargos geprüft.

Warum ist eine nXt-Lösung für diese Aufgabenstellung besonders gut geeignet?

Sanktionslistenprüfungen, die im Wesentlichen auf der Ähnlichkeit von Adressen basieren, lassen sich weitgehend maschinell automatisieren. Es bleibt aber immer ein kleiner Anteil an Grenzfällen, bei denen Algorithmen eine Übereinstimmung der Adressen nicht zweifelsfrei erkennen können. Beispielsweise bei unterschiedlichen Schreibweisen von Namen, bei Tippfehlern oder auch, wenn auf den Sanktionslisten nur wenige Erkennungsmerkmale eingetragen sind (z. B. nur ein Name, aber keine Ortsangabe). Daher muss geschultes Fachpersonal die finale Beurteilung und Entscheidung übernehmen, ob eine Adresse wirklich einem Sanktionslisteneintrag entspricht, und wenn ja, wie damit weiter zu verfahren ist.

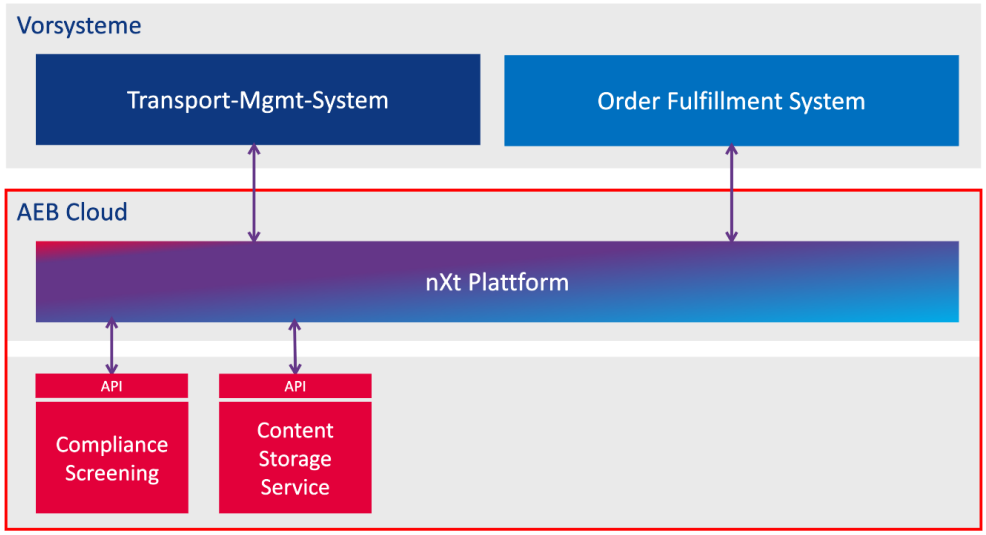
In produzierenden Industrieunternehmen treten diese Fälle relativ selten auf, sodass dieser Ausnahmeprozess i. d. R. nicht detailliert IT-gestützt abgebildet werden muss. Es genügt meist, den Compliance-Beauftragten darüber umgehend zu informieren und den auslösenden Beleg bis zur Klärung vorübergehend zu sperren.

Bei den sehr hohen Sendungsvolumina und sehr vielfältiger und dynamischer Adressdatenstruktur in unserem Szenario fallen hier allerdings so viele dieser Grenzfälle an, dass für eine erste Vorprüfung dieser Grenzfälle ein externer Dienstleister eingesetzt wird und insgesamt ein besonders effizienter und mehrstufiger Prüf- und Freigabeprozess erforderlich ist. Die Einhaltung der Sorgfaltspflicht muss im Falle einer Prüfung durch Zollbehörden gerade auch in solch einem mehrstufigen Prozess nachgewiesen werden können. Auch im Zuge einer Wirtschaftsprüfung kann dies erforderlich sein. Entsprechende Prüfpfade (sogenannte Audit-Trails) müssen also auf den hier definierten Prüf- und Freigabeprozess ausgelegt sein und für spätere Prüfzwecke revisionssicher aufbewahrt und archiviert werden.

Dieser Freigabeprozess wurde als nXt-Lösung in der AEB Cloud modelliert und realisiert. Dabei steht einerseits die Transparenz und Konsistenz des mehrstufigen Prüf- und Freigabeprozesses im Mittelpunkt, aber auch die Effizienz in der Bearbeitung der konkreten Prüfvorgänge. Diese wird neben einem effizienten Prozessablauf auch durch eine auf die konkreten Benutzer und ihre Aufgaben ausgerichtete Gestaltung der Benutzerdialoge (User-Tasks, Apps) erreicht (siehe auch 1-1-3-Prinzip).

Welche AEB Business Services kommen zum Einsatz?

- Prüfung der Geschäftspartner gegen globale Sanktionslisten oder geltende Embargos -> Compliance Screening
- Langzeitdatenspeicherung und Archivierung -> Content Storage Service



3.3 Szenario 3: Versandlogistik

Dritter Beispielkunde ist ein führender europäischer Hersteller von Abdichtsystemen für Kabel, Rohre und Hauseinführungen und beliefert täglich eine Vielzahl unterschiedlicher Kunden, wie z. B. Energieversorger, Stadtwerke, Bauunternehmen, sowie die Industrie.

Die Einhaltung von – meist recht kurzen – bedarfsgerechten Lieferzeiten, oft direkt zum Ort der Verwendung, zählen zum Erfolgsrezept des mittelständischen Unternehmens. Verschiedene Materialien aus Fertigung oder Konfektionierung sowie Lagerartikel werden in der Intralogistik zum gemeinsamen Versand an den Kunden zusammengeführt.

Mit SAP® S/4HANA und SAP® EWM setzt der Hersteller moderne Systeme ein, die er on-premise (also im eigenen Haus) betreibt. Für die Abwicklung der Versandlogistik hat er sich für eine ergänzende Lösung in der AEB Cloud auf Basis der nXt-Plattform entschieden.

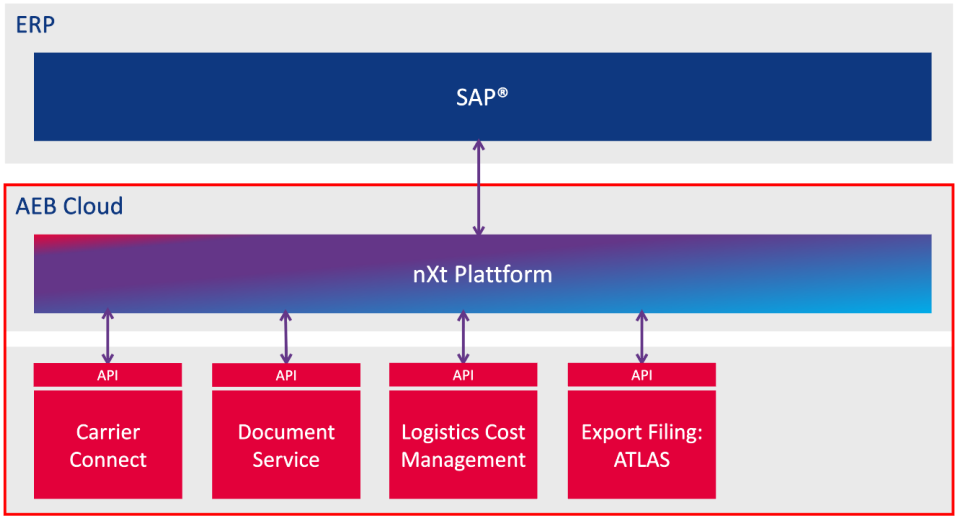
Warum ist eine nXt-Lösung für diese Aufgabenstellung besonders gut geeignet?

- Wichtig ist dem Unternehmen eine Entkopplung von den zentralen SAP-Systemen des Unternehmens. Separation of concerns, mehr Flexibilität in der Anwendung und eine einfachere Anpassbarkeit an die Anforderungen der Logistik werden als Vorteil gesehen.

- Heute erfolgt der Versand der Waren von einem zentralen Standort aus. Zukünftig sollen auch externe Lager sowie externe Lohnbearbeiter Waren direkt versenden und dabei das AEB-Versandsystem nutzen. Der einfache Zugriff auf die Cloud Lösung über das Internet ist dabei eine wichtige Voraussetzung.
- Systemfreigaben, die Ansteuerung von Label-Druckern oder der Anschluss von anderer lokaler Hardware (Scanner, Waagen etc.) sind mit AEB Business Services und nXt-Templates deutlich weniger aufwändig als in Eigenentwicklung.
- In den Versandlogistikprozess wurden diverse Maßnahmen zur Qualitätssicherung integriert. Über die Prozesssteuerung der AEB-Plattform wird dies transparent und flexibel abgebildet. Auf den Kontext optimierte, schlanke Apps bauen auf dem Prozess auf.
- So wird z. B. über Business Rules bei ausgewählten Kunden das 4-Augen-Prinzip beim Verpacken von Lieferungen angewendet. Die Versandleitung erhält dann über den Workflow gesteuert eine Aufgabe zugeteilt. Erst nach Freigabe der Versandleitung in der zugehörigen nXt-App kann die Ware fertig verpackt und versendet werden.
- Umstrukturierungen und Neuorganisation (hier getrieben durch räumliche und bauliche Veränderungen) führten zu neuen Anforderungen an die Logistik. Für die Zusammenführung der Waren in einer neu eingezogenen Pufferzone konzipierte AEB zusammen mit den Key Usern eine neue optimierte App. Durch das auf UI-Komponenten basierte Framework der AEB erfolgte die Umsetzung der App in nur wenigen Tagen, sie wird auf einem handelsüblichen Touch Pad betrieben.

Welche AEB Business Services kommen zum Einsatz?

- Aufbereitung von TDL-Labels und EDI-Nachrichten -> Carrier Connect
- Aufbereitung von kundenindividuellen Labels und Dokumenten -> Document Service
- Kalkulation und Kontrolle von Transportkosten -> Logistics Cost Management
- Ausfuhranmeldung (ATLAS) an den Zoll und Druck des Ausfuhrbegleitdokuments -> Export Filing: ATLAS



AEB SE . Hauptsitz . Sigmaringer Straße 109 . 70567 Stuttgart . Deutschland . +49 711 72842 0 . www.aeb.com . info.de@aeb.com . Registergericht: Amtsgericht Stuttgart . HRB 767 414 . Geschäftsführende Direktoren: Matthias Kieß, Markus Meißner . Vorsitzende des Verwaltungsrats: Maria Meißner

Standorte
Düsseldorf . Hamburg . Lübeck . Mainz . Malmö . Manila . München . New York . Prag . Rotterdam . Salzburg . Singapur . Soest . Stuttgart . Warwick . Zürich