



Schlussbericht zum Forschungsprojekt

# **Port Energy Management Dashboard**

## Digitale Leitwarte zur Analyse und Steuerung von Energieflüssen in Häfen

### **dashPORT**

Patrick Bruns, M. Sc.

Lasse Hammer, M. Sc.

Daniel Lange, M. Sc.

Prof. Dr. Sebastian Lehnhoff

Gefördert durch:



Bundesministerium  
für Digitales  
und Verkehr

aufgrund eines Beschlusses  
des Deutschen Bundestages

Das diesem Bericht zugrundeliegende  
Vorhaben wurde mit Mitteln des  
Bundesministerium für Digitales und Verkehr  
unter dem Förderkennzeichen 19H19005B  
gefördert.

Die Verantwortung für den Inhalt dieser  
Veröffentlichung liegt bei den Autoren.

Oldenburg, 20.03.2023

Projektverbund:

„dashPORT - Port Energy Management Dashboard Digitale Leitwarte zur Analyse und Steuerung von Energieflüssen in Häfen“

Verbundkoordinator:

Niedersachsen Ports GmbH & Co. KG  
Hindenburgstr. 26-30  
26122 Oldenburg

Kontakt:

Dr. Matthäus Wuczkowski  
Telefon: 0441-35020-613  
E-Mail: mwuczkowski@nports.de

Verbundpartner:

Fraunhofer-Center für Maritime Logistik und Dienstleistungen CML  
Am Schwarzenberg-Campus 4, Gebäude D  
21073 Hamburg

OFFIS – Institut für Informatik  
Escherweg 2  
26121 Oldenburg

J. Müller AG  
Neustadtstraße 15  
26919 Brake

# Inhalt

1 Projektbeschreibung .....	4
1.1 Aufgabenstellung.....	4
1.2 Voraussetzungen.....	4
1.3 Planung und Ablauf des Vorhabens .....	5
1.4. Stand der Technik und Wissenschaft.....	6
1.5 Kooperation .....	7
2 Projektergebnisse .....	8
2.1 Partnerspezifischer Beitrag.....	8
2.2.1 AP 1: Bestandsaufnahme/ IST-Analyse .....	8
2.2.2 AP 2: Analyse und Konzeption .....	9
2.2.3 AP 3: Umsetzung notwendiger Infrastrukturmaßnahmen für dashPORT .....	9
2.2.4 AP 4: Datenanalyse .....	10
2.2.4.1 Übersicht erfasster Daten .....	10
2.2.4.2 Grobkonzept Prognosemodelle .....	10
2.2.4.3 Eingabedaten .....	11
2.2.4.4 Datenvorverarbeitung .....	12
2.2.4.5 Prognosemodelle.....	12
2.2.4.6 Prognosemodelle für Handlungsempfehlungen .....	13
2.2.5 (TAP 6.1) Konzeptionierung dashPORT.....	13
2.2.5.1 Anforderungserhebung .....	13
2.2.5.2 Technische Beschreibung des Gesamtsystems .....	15
2.2.5.3 Entwurf der grafischen Benutzeroberfläche .....	18
2.2.5.4 Entwurf des Rollenkonzepts .....	24
2.2.6 (TAP 6.2) Implementierung dashPORT.....	24
2.2.3.2 Datenbanksystem (InfluxDB, Telegraf) .....	24
2.2.3.1 Datenstrommanagementsystem (Flink, RabbitMQ) .....	25
2.2.6.4 Umsetzung der grafischen Benutzeroberfläche .....	28
2.2.7 (TAP 6.3) Tests dashPORT .....	33
2.2.8 AP 7: Feldtest im Hafen .....	33
2.2.9 AP 8: Evaluierung und Verwertung .....	34
2.3 Verwendung der Zuwendung.....	34
2.4 Notwendigkeit der geleisteten Arbeit.....	34
2.5 Verwertbarkeit .....	35
2.6 Fortschritt anderer Stellen .....	35
2.7 Veröffentlichungen.....	36
3 Abbildungsverzeichnis .....	36
4 Tabellenverzeichnis .....	38

# 1 Projektbeschreibung

Sowohl der Infrastrukturbetreiber NPorts als auch das Umschlagsunternehmen J. Müller haben für ihren Standort Hafen Brake unabhängig voneinander Energiemanagementsysteme eingeführt. Somit erfolgt auch die Erfassung von Verbrauchswerten gemessener Elektrizitätsverbraucher unabhängig voneinander und teilweise noch mit manueller Ablesung nicht flächendeckend. Eine Berücksichtigung von Wechselwirkungen ist daher derzeit nicht möglich. Das Projekt DashPORT zielt daher auf den Gesamthafen Brake. Es sollen wesentliche Stromverbrauchsdaten des Infrastrukturbetreibers und des ansässigen Umschlagsunternehmens flächendeckend erhoben, zusammengeführt, visualisiert und prognostiziert werden, um ein intelligentes Energie- und Flexibilitätsmanagement zu realisieren. Kern des Projekts ist der Software-Prototyp DashPORT, welcher historische, aktuelle und prognostizierte Verbräuche visualisiert und konkrete Maßnahmen vorschlägt, um teure Lastspitzen proaktiv zu verhindern. Im Rahmen des Projekts werden die Prozesse des Hafens und die Schnittstellen der beiden Hafenakteure aus energetischer Sicht analysiert, Effizienzmaßnahmen abgeleitet und Handlungsempfehlungen entwickelt, welche in die dashPORT-Software integriert werden.

## 1.1 Aufgabenstellung

Um systematisch Energieeffizienz- und Flexibilisierungspotenziale sichtbar und nutzbar zu machen, ist das konkrete Ziel des Vorhabens die prototypische Entwicklung und pilothafte Anwendung des Port Energy Management Werkzeugs DashPORT mit folgenden Eigenschaften:

- Datenstrommanagement und Visualisierung von Stromverbrauchsdaten (historisch und aktuell)
- Prognose von zukünftigen Lastverläufen unter Einbezug von Daten über geplante Schiffsankünfte
- Einstufung der Stromverbrauchsdaten anhand von Benchmark-Werten
- Intelligentes Warnsystem für extreme und unnötige Verbräuche
- Umsetzung von Energieeffizienzmaßnahmen und energetischen Prozessoptimierungen im Hafen
- Flexible Steuerung von wesentlichen Verbrauchern in Bezug auf Energieeffizienz (z.B. Beleuchtung)

Dazu soll

- Die vorhandene Messinfrastruktur ergänzt werden, um den Leistungsbedarf aller wesentlichen Verbraucher zu erfassen
- Eine visuelle Aufbereitung der Energieverbräuche des gesamten Hafens erfolgen, um eine ganzheitliche Sicht auf Verbräuche über den gesamten Hafen zu erhalten
- Ein Warnsystem für extreme und unnötige (prognostizierte) Verbräuche integriert werden
- Das System Handlungsempfehlungen generieren, mit konkreten und auf Prozesswissen basierenden Vorschlägen von Verbrauchern, deren Verbrauch reduziert werden könnte

## 1.2 Voraussetzungen

Das hier beschriebene Projekt dashPORT ist ein anwendungsorientiertes Forschungs- und Entwicklungsprojekt gemäß der Förderrichtlinie Innovative Hafentechnologien (IHATEC) des

Bundesministeriums für Verkehr und digitale Infrastruktur (BMVI). Der zweite Förderaufruf vom 15.09.2017 nennt Zuwendungszwecke, die von dashPORT wie folgt adressiert werden:

Digitale Infrastruktur verbessern und die stärkere Nutzung von der IT in den Häfen vorantreiben

DashPORT zielt auf eine vollständige Digitalisierung der Erfassung und Überwachung der Energieflüsse im Hafen. Durch die flächendeckende Ausstattung aller relevanten Verbraucher des Infrastruktur- und Umschlagsbereichs mit digitalen Messeinrichtungen und der digitalen Übertragung der Messdaten werden die Energieflüsse des Hafens kontinuierlich sichtbar gemacht und kurzfristige Maßnahmen zur Verbrauchsreduktion ermöglicht. Die erstmalige Zusammenführung der Energieverbrauchsdaten von zwei Hafenakteuren in einem gemeinsamen IT-System ermöglicht es, Wechselwirkungen zu erkennen und Prognosen für zukünftige Verbräuche und Verbrauchsspitzen zu erstellen.

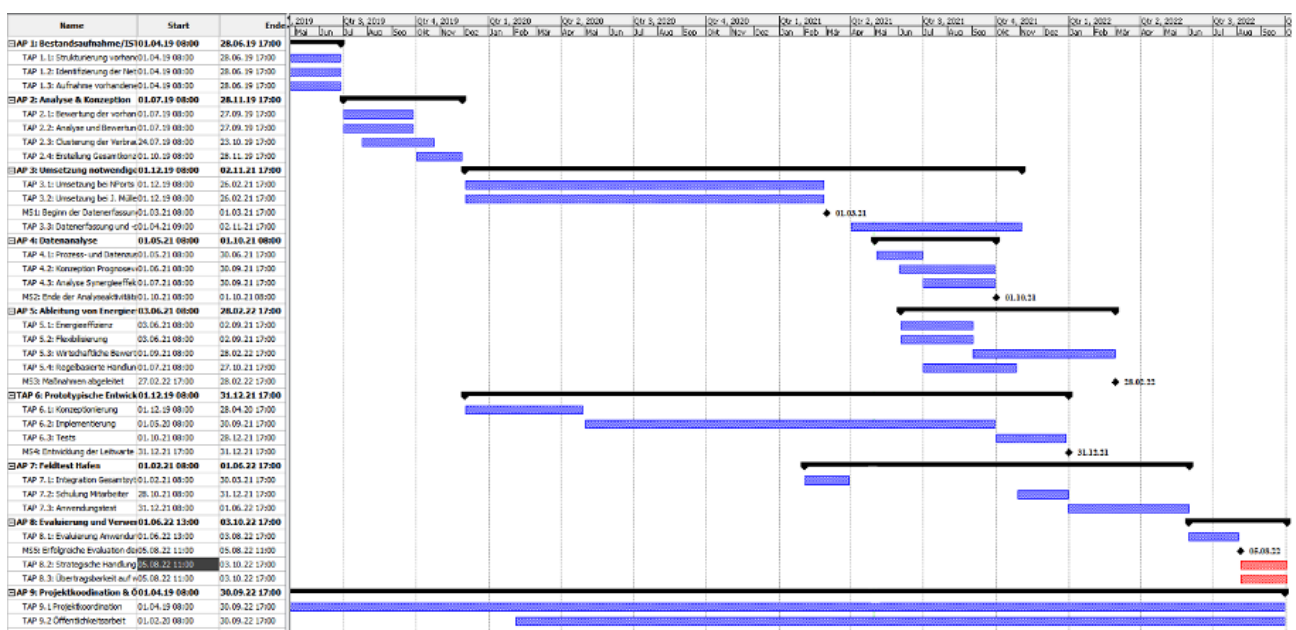
Entwicklung innovativer Hafentechnologien fördern, die zur Verbesserung des Umwelt- und Klimaschutzes beitragen

dashPORT analysiert die Hafenprozesse aus energetischer Sicht übergreifend über die beiden Hafenakteure Infrastrukturbetreiber und Umschlagunternehmen hinweg und leitet daraus Effizienzmaßnahmen und Handlungsmaßnahmen im Bereich Stromverbrauch ab. Durch die Visualisierung der Stromverbrauchsdaten und deren Wechselwirkungen sollen unnötige Verbräuche vermieden werden und Einsparpotentiale intuitiv angezeigt werden. Die Prognosefähigkeit von dashPORT ermöglicht es auch, präventiv Maßnahmen zu ergreifen und so zur Verbesserung des Umwelt- und Klimaschutzes beizutragen.

Das Projektvorhaben dashPORT ist damit dem in Nr. 4.2 der IHATEC-Förderrichtlinie genannten Schwerpunkt g) Technische Innovationen zur Steigerung der Energieeffizienz im Hafen und Verringerung der Umweltbelastung zuzuordnen.

## 1.3 Planung und Ablauf des Vorhabens

Die Abstimmung mit den Projektpartnern verlief zielführend und kommunikativ. In den regelmäßigen monatlichen (Online)-Meetings konnte der Arbeitsstand ausgetauscht und die Planung umgesetzt werden.



#### Abbildung 1: Gantt Chart dashPORT

Der Einbau der notwendigen Messinfrastruktur hat sich aufgrund der Pandemiesituation schwierig gestaltet: Geltende Zugangsbeschränkungen im Hafengebiet und eine geringe Verfügbarkeit von entsprechenden Dienstleistern verzögerten die Infrastrukturmaßnahmen erheblich.

Um diese Verzögerungen abzufangen, wurden Arbeitsschritte soweit möglich vorgezogen, insbesondere die Konzeption und das Design der späteren Benutzeroberfläche. Dennoch war es für das Erreichen der Projektziele notwendig, die Laufzeit des Projektes um 6 Monate zu verlängern, um genügend Daten für das Training des KI-Prognosemodells und die Entwicklung des Entscheidungsunterstützungssystems zu sammeln. Die alternative Extrapolation der Daten einzelner Monate war nicht zielführend, da Saisonalitäten und Trends so nicht erkannt hätten werden können.

Entsprechend war eine kostenneutralen Laufzeitverlängerung notwendig, durch die sich das Projektende vom 31.03.2022 um 6 Monate auf den 30.09.2022 verschob.

## 1.4. Stand der Technik und Wissenschaft

Das erwartete Endergebnis des geplanten Vorhabens ist ein von einem Terminal Operating System (TOS) unabhängiger Prototyp zur Kontrolle, Prognose und Steuerung inkl. visueller Aufbereitung von Verbrauchswerten technischer Anlagen in Häfen. Dieser Prototyp soll auch und gerade für kleinere Häfen einsetzbar sein, deren Digitalisierungsgrad und Prozessreife üblicherweise nicht so hoch ist wie es bei großen Containerterminals der Fall ist. Einsortieren lässt sich das Vorhaben in die aktuelle (wissenschaftliche) Diskussion der zunehmenden Anzahl von Smart Ports. Heilig et al. (2017) analysieren die digitale Transformation von Häfen. Die Autoren beschreiben IT-Lösungen als wesentlich, um mit den zukünftigen Herausforderungen in Häfen umzugehen. In der Veröffentlichung genannte Beispiele für Smart Ports sind u. a. der Großhafen Rotterdam oder der Containerhafen Hamburg.<sup>1</sup> Bei genauerer Betrachtung dieser Häfen fällt auf, dass die Themen Energie und Energiemanagement wesentliche Bestandteile der Smart Port Strategie sind (z. B. smartPORT Energy in Hamburg und Smart Energy and Industry in Rotterdam). Aktuelle Veröffentlichungen von Accenture<sup>2</sup> und Deloitte<sup>3</sup> bestätigen dies. Tabelle 1 grenzt das Vorhaben dashPORT von relevanten laufenden und bereits abgeschlossenen Projekten ab.

Akronym	Inhalt	Abgrenzung des Projektvorhabens
GreenCranes, (TEN-T; 08/2012-06/2014)	Einführung der ISO 50001 und Echtzeit Monitoring ausgewählter Verbraucher im CT Koper (SLO)	dashPORT betrachtet kleinere Multipurpose-Terminals und ist TOS-unabhängig konzipiert.
SEAMS in Sea Terminals (TEN-T; 03/2014-12/2015)	Einführung Smart Energy-Efficient und Adaptive Management System im CT Valencia; Live Daten durch Datenaufzeichnungsstation im	dashPORT betrachtet kleinere Multipurpose-Terminals und ist TOS-unabhängig konzipiert.

<sup>1</sup> <https://scholarspace.manoa.hawaii.edu/bitstream/10125/41313/1/paper0164.pdf>

<sup>2</sup> [https://www.accenture.com/t20161012T003018Z\\_w\\_us-en\\_acnmedia/PDF-29/accenture-connected-ports-driving-future-trade.pdf](https://www.accenture.com/t20161012T003018Z_w_us-en_acnmedia/PDF-29/accenture-connected-ports-driving-future-trade.pdf)

<sup>3</sup> <https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/energy-resources/deloitte-nl-er-port-services-smart-ports.pdf>

	Terminalfahrzeug (Black Box); Rückmeldung ausgewählter Fahrzeug-parameter; Dynamische Ausleuchtung des Terminals	
SustEnergyPorts (IHATEC; 07/2017-07/2019)	Ökologische und ökonomische Bewertung ausgewählter Maßnahmen zur Erhöhung der Energieeffizienz. Hierfür soll sowohl der Gesamtenergieverbrauch in Form einer Ökobilanz, als auch ökonomische Auswirkungen durch Simulation der Anwendung der Maßnahmen in einem Modell betrachtet werden. Die zu bewertenden Maßnahmen werden derzeit ausgewählt.	dashPORT soll für den operativen Betrieb ein Echtzeit-System zur Erfassung, Speicherung und Bewertung der tatsächlichen Verbrauchswerte konkreter Anlagen entwickeln. Es wird erwartet, dass dadurch nicht nur ungerechtfertigt hohe Verbräuche ersichtlich werden, sondern auch Verbrauchsinformationen z. B. für SmartGrids bereitgestellt werden können.
BESIC (IKT für Elektromobilität II)	Nutzung von Flexibilitätspotential von automatisierten batterie-elektrischen Containertransportern (AGVs) für verschiedene Anwendungsfälle wie zum Beispiel Lastgangoptimierung und Bereitstellung von Minutenreserveleistung. Entwicklung eines Simulationsmodells des Hafens mit Lastgangprognosefähigkeit.	dashPORT betrachtet kleinere Multipurpose-Terminals, Elektromobilität ist derzeit noch kein wesentlicher Energieverbraucher in dem betrachteten Hafen.

**Tabelle 1: Abgrenzung von verwandten Arbeiten**

Bezogen sich bisher erfolgte Ansätze des Energiemanagements in europäischen Häfen respektive der Live-Datensammlung und -auswertung zum überwiegenden Teil auf Containerterminals, soll der Fokus mit dem Projektvorhaben dashPORT auf hafenübergreifende Energiemanagementsysteme in Multi-Purpose Terminals gelenkt werden. Diese waren bisher von einem nachrangigen Interesse, da Containerterminals durch die hohe Anzahl an standardisierten Vorgängen und Terminalequipment mehr Optimierungsansätze boten. dashPORT soll vor allem die Belange von kleineren bis mittleren Multipurpose-Häfen wie Brake hinsichtlich ihres Energiemanagements berücksichtigen und eine Gesamtsicht auf alle Verbraucher eines solchen Hafens rollenbasiert ermöglichen.

## 1.5 Kooperation

dashPORT wurde in Zusammenarbeit aller Projektpartner durchgeführt. Zum Konsortium zählen neben OFFIS und dem Fraunhofer CML die Nutzer des dashPORTs, die Hafenumschlaggesellschaft Niedersachsen Ports GmbH und die Terminalbetreiberin J.Müller AG.

Zu Besprechungen und zur Bewertung der Zwischenergebnisse im laufenden Projekt sowie zur Organisation innerhalb des Vorhabens wurden monatliche Videokonferenzen unter Teilnahme der Projektpartner durchgeführt. Bei Bedarf wurden außerdem persönliche Treffen (Vor COVID Pandemie) und virtuelle Treffen (während COVID Pandemie) mit den jeweiligen Projektpartnern zum fachlichen Austausch der Projektarbeit durchgeführt. Für die effektive Zusammenarbeit zwischen den Beteiligten wurde vom Fraunhofer CML ein cloudbasierter Datenraum zur Verfügung gestellt, welcher das Projektmanagement auf Distanz hinsichtlich des Austausches, der

Verwaltung und der Ablage von Daten, Zwischenergebnissen, Berichten und administrativen Unterlagen wesentlich vereinfacht hat.

## 2 Projektergebnisse

### 2.1 Partnerspezifischer Beitrag

Im Rahmen des partnerspezifischen Abschlussberichts wird im Folgenden hauptsächlich auf die durch das OFFIS entwickelten Konzepte und deren Umsetzung eingegangen.

#### 2.2.1 AP 1: Bestandsaufnahme/ IST-Analyse

Die Bestandsaufnahme wurde vom Fraunhofer CML geleitet. In den Teilarbeitspaketen 1.1 und 1.2 hat OFFIS durch Teilnahme an den entsprechenden Terminen und bei technischen Fragestellungen unterstützt. Die technische Bestandsaufnahme in Teilarbeitspaket 1.3 wurde von OFFIS durchgeführt.

Die Grundlage für die Implementierung bildet der Systemkontext und die Rahmenbedingungen in denen die digitale Leitwarte agieren wird. Dafür wurden ein AP 1 Termine zur Bestandsaufnahme mit den Projektpartnern durchgeführt. Auf Basis der Ergebnisse aus den Terminen wurde dann eine Systemkontextanalyse durchgeführt.

Bei der Systemkontextanalyse ergab sich der in Abbildung 2 dargestellte Kontext. In gemeinsame Workshops wurde festgestellt, dass Messdaten von J.Müller über einen OPC-Server abgerufen werden können. Bei der Messdatenübertragung von NPorts wurde sich dafür entschieden die IEC 60870-5-104 Schnittstelle zu verwenden. Bei beiden Anwender-Unternehmen gibt es verschiedene Datenquellen für Prozessinformationen. Als weitere wichtige Datenquellen für die dashPORT-Plattform wurde die webbasierte Schiffsliste von J. Müller und der AIS-Empfänger von NPorts identifiziert. Beide Datenquellen haben Potential die Prognose-Güte und Handlungsempfehlungen zu verbessern, indem sie Informationen über aktuelle Prozesse im Hafen weitergeben.

Des Weiteren gibt es Anwender bei beiden Unternehmen, die sowohl über Arbeitsplatzrechner als auch mobilen Geräten auf das Dashboard zugreifen müssen. Daher wurde sich dafür entschieden die Plattform als Web-Applikation bereitzustellen.

Die dashPORT-Plattform wird für die Projektlaufzeit auf virtuellen Maschinen auf Servern von J.Müller betrieben. Um die Datenübertragung und Zugriff von NPorts auf das System zu gewährleisten wurden daher entsprechende Netzwerk-Verbindungen zum virtuellen Netzwerk von dashPORT bei J.Müller hergestellt.



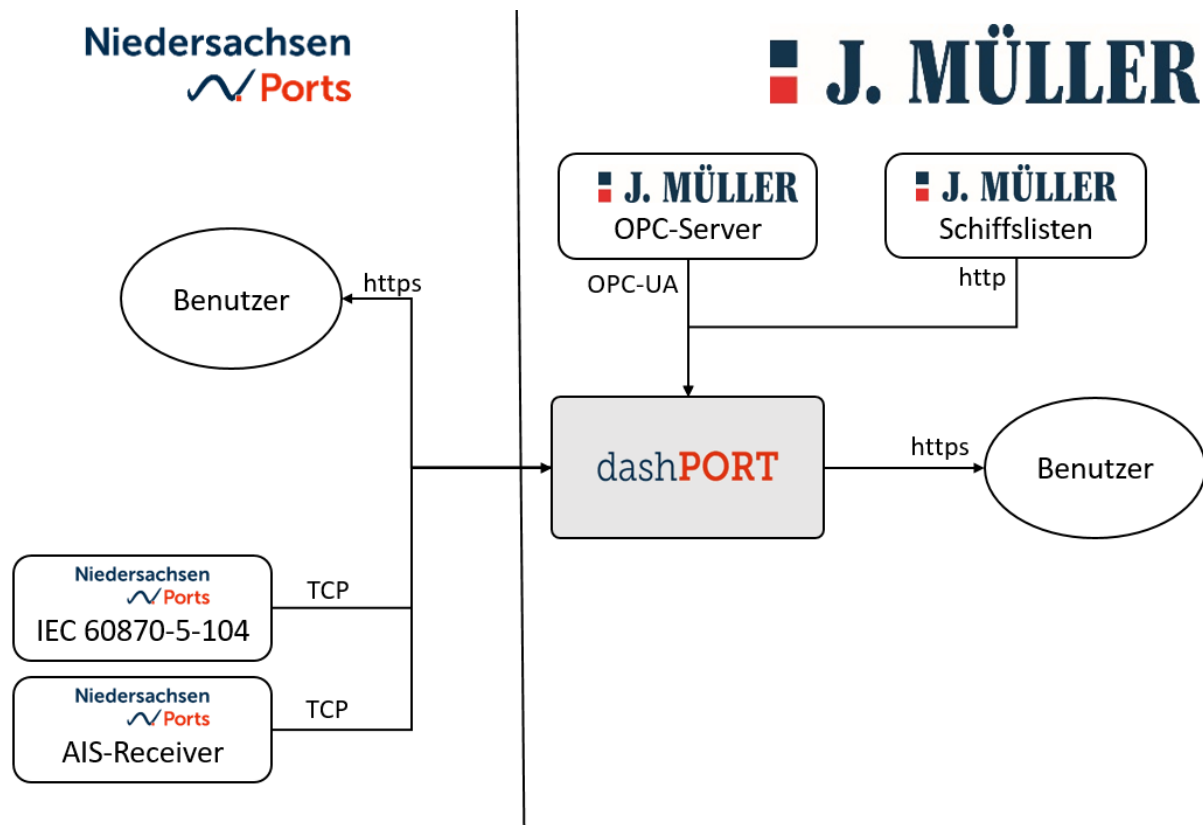


Abbildung 2: Systemkontextsicht auf dashPORT

### 2.2.2 AP 2: Analyse und Konzeption

Die Analyse und Konzeption wurde vom Fraunhofer CML geleitet. OFFIS hat durch die Teilnahme an den entsprechenden Terminen und bei technischen Fragestellungen unterstützt.

### 2.2.3 AP 3: Umsetzung notwendiger Infrastrukturmaßnahmen für dashPORT

Die Umsetzung der Infrastrukturmaßnahmen in den Teilarbeitspaketen 3.1 und 3.2 wurden von den Industriepartnern NPorts und J. Müller geleitet. OFFIS stand hier für technische Vorabstimmungen für die spätere Datenanbindung unterstützend zur Seite.

In Teilarbeitspaket 3.3 wurde die Datenerfassung und –speicherung durchgeführt. OFFIS hat dabei regelmäßig die ordnungsgemäße Erfassung und Speicherung überprüft, sowie die Daten inhaltlich überprüft. Nachdem die Datenverbindung hergestellt wurde, mussten zunächst die übertragenen Daten auf Vollständigkeit und Plausibilität untersucht werden. Dabei wurden Datenpunkte aufgedeckt, die Fehler aufwiesen, wie beispielsweise negative Leistungswerte, falsche Wertebereiche, dauerhafte Übertragung von 0 Werten an Messpunkten, wo dies nicht korrekt ist und gänzlich fehlende Datenübertragung. In diversen Überprüfungen wurden diese Fehler aufgedeckt, an die Partner übermittelt und in der Datenerfassungsinfrastruktur behoben. In einem abschließenden, mehrtägigen Workshop wurden alle Messpunkte noch einmal überprüft und abgeglichen, sodass im Anschluss mit der Datenerfassung der bereinigten Daten für das Training der Machine Learning Modelle begonnen werden konnte.

## 2.2.4 AP 4: Datenanalyse

Die Datenanalyse wurde fhrend durch Fraunhofer CML durchgefhrt. OFFIS hat Teilarbeitspaket 4.1 bei der Datenzusammenfhrung technisch untersttzt. Fhrend war OFFIS in Teilarbeitspaket 4.2 ttig, in dem die Prognosemodule konzipiert wurden.

### 2.2.4.1 bersicht erfasster Daten

Fr die beiden Projektpartner J. Mller und NPorts werden an jeder installierten Zhlerstelle die momentan abgerufene Leistung in kW, sowie der Zhlerstand in kWh bermittelt. NPorts bertrgt alle 15 Sekunden die Werte fr alle Zhlstellen. Bei J. Mller werden nur dann Werte bertragen, wenn sich eine nderung ergeben hat.

Diese Werte an in die Datenbank bertragen und dort separat fr jede Zhlerstelle die aktuelle Leistung und der Zhlerstand gespeichert. Die Daten werden nach Partnern getrennt abgelegt, sodass der jeweilige Partner nur auf seine Daten Zugriff hat um den Datenschutz zu gewhrleisten.

In den folgenden Tabellen sind Beispielauszge fr die Leistungswerte und Zhlerstnde des Zhlers AH\_ES1\_J01 der Firma NPorts und P\_MS\_001 der Firma J. Mller dargestellt.

_time	_value	_measurement	company	meterId
2021-10-14 T22:34:52.820Z	17.27224586378549	smartmeter_nports	nports	AH_ES1_J01_P

Tabelle 2: Beispielauszug Leistungswerte AH\_ES1\_J01

_time	_value	_measurement	company	meterId
2021-10-14 T22:34:52.820Z	1149442.338164251 3	smartmeter_nports	nports	AH_ES1_J01_W

Tabelle 3: Beispielauszug Zhlerstand AH\_ES1\_J01

_time	_value	_measurement	company	meterId
2021-10-14 T22:34:52.820Z	166.40053258524551	smartmeter_jmueller	jmueller	P_MS_001_P

Tabelle 4: Beispielauszug Leistungswerte P\_MS\_001

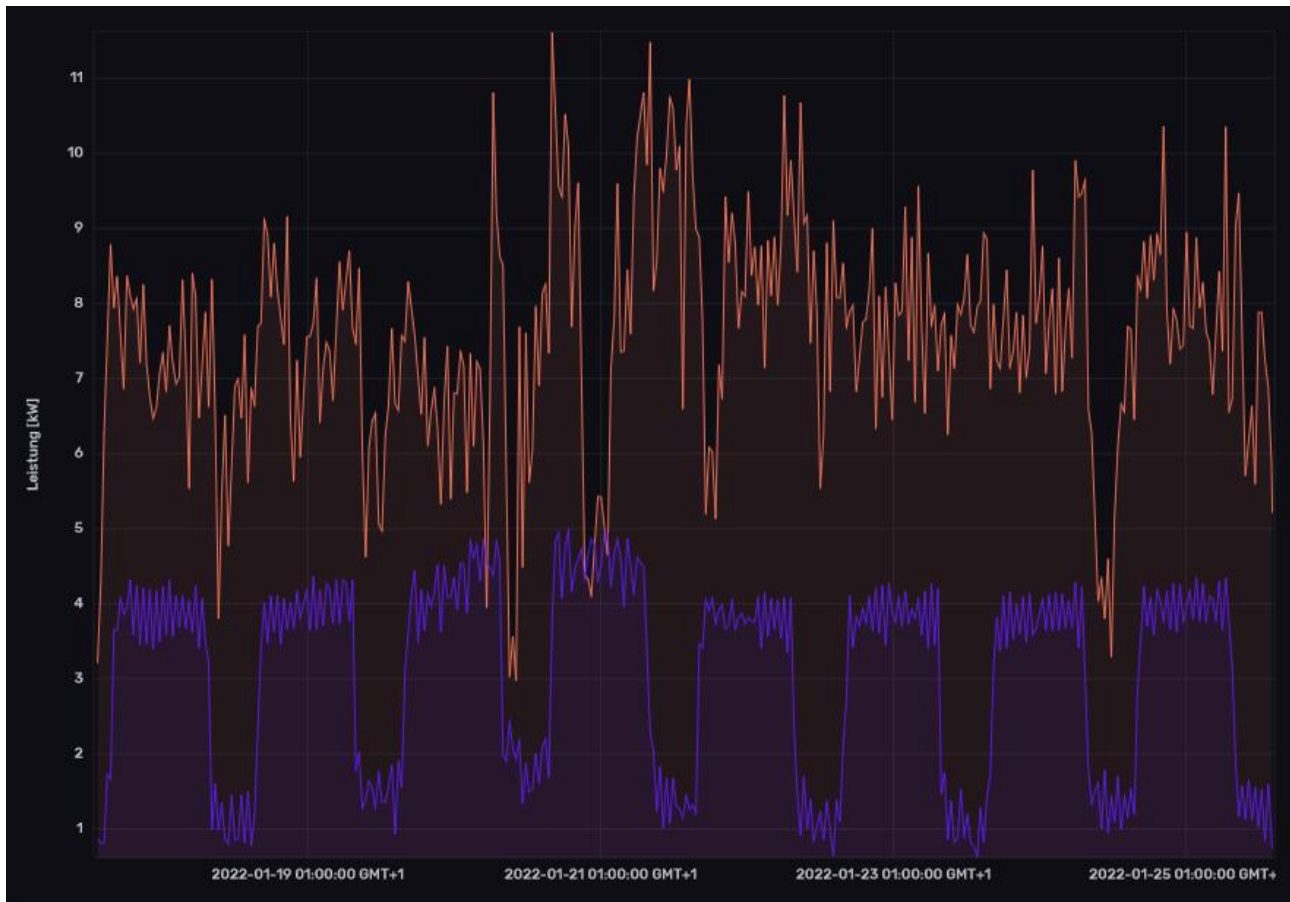
_time	_value	_measurement	company	meterId
2021-10-14 T22:34:52.820Z	2840394.958481613	smartmeter_jmueller	jmueller	P_MS_001_W

Tabelle 5: Beispielauszug Zhlerstnde P\_MS\_001

### 2.2.4.2 Grobkonzept Prognosemodelle

Um eine mgliche berschreitung der Lastspitze in Zukunft zu prognostizieren muss der Lastverlauf der Einspeisungen prognostiziert werden. Fr jede Einspeisung werden die Lastspitzen separat abgerechnet. Das bedeutet, dass die mgliche Lastspitzenberschreitung fr jede Einspeisung einzeln prognostiziert und berprft werden muss. Da sich die Lastverlufe je nach Einspeisung sowohl im Wertebereich, als auch im Verlauf unterscheiden, wird fr jede Lastspitze

ein eigenes Prognosemodell trainiert, welches für die Vorhersagen im Gesamtsystem genutzt wird. Die unterschiedlichen Lastverläufe sind beispielhaft in Abbildung 3 für die Einspeisungen AH\_EZ1\_J01 und AH\_HV11\_V3\_P1 der Firma NPorts im Zeitraum vom 17.01.2022 bis 25.01.2022 dargestellt.



**Abbildung 3: Gegenüberstellung der Einspeisungen AH\_EZ1\_J01 (orange) und AH\_HV11\_V3\_P1 (blau)**

Lastspitzen werden vom Netzbetreiber in festen 15 Minuten Intervallen erhoben. Diese Intervalle gehen für jede Stunde von der vollen Stunde aus, es handelt sich also nicht um einen gleitenden Durchschnitt. Die abgerufene Leistung wird in diesen 15 Minuten Intervallen gemittelt und daraus wird die Lastspitze berechnet. Sehr kurze Ausschläge, beispielsweise wenige Sekunden, bedeuten also nicht direkt eine für die Abrechnung relevante Lastspitze. Aus diesem Grund ist es sinnvoll, die Prognose auch für diese 15-Minuten Intervalle durchzuführen und nicht, auf 15-Sekunden Basis, auf der die Daten erfasst werden. Die Ausgabe der Prognosemodelle sind deshalb die prognostizierten Leistungswerte der Einspeisung gemittelt auf die 15-Minuten Intervalle der nächsten Stunde.

### 2.2.4.3 Eingabedaten

Die wichtigsten Eingabedaten, aus denen die Prognose berechnet wird, sind die durch die neu installierten Zähler erfassten Leistungsdaten. Genauer gesagt die Leistungsdaten der jeweiligen Einspeisung, für die die Prognose erstellt werden soll. Da alle Einspeisungen direkt mit eigenen Zählern versorgt wurden, ist eine Aggregation der untergeordneten Zählerwerte nicht nötig und es kann direkt am Einspeisepunkt die tatsächliche Einspeisung abgelesen werden. Diese enthält dann sowohl die tatsächlich in Anspruch genommenen Leistungen an den Abnahmestellen und alle Systemverluste. Da die Ausgabe in 15-Minuten Intervallen erfolgt, werden auch die Leistungsdaten in diesem Format eingegeben. Dazu werden die Werte der letzten Stunde auf die vier letzten Intervalle aggregiert und in die Modelle eingegeben.

Zusätzlich könnten Prozessinformationen die Prognosegüte verbessern. Da große Teile der Verbraucher prozess- und umschlagsabhängig sind wäre es sinnvoll diese Daten einzubinden. Beispielsweise existiert beim Partner J. Müller eine sogenannte Schiffsliste, in der die geplanten Schiffsankünfte erfasst werden. Die Liste ist zwar digital auf einer Website verfügbar, allerdings werden die Felder händisch gepflegt, was dazu führt, dass es keine einheitlichen Bezeichnungen, Zeitangaben und Formatierungen gibt. Dadurch war es im Projektrahmen nicht möglich, die vorhandene Schiffsliste in ein maschinenlesbares Format zu überführen und damit für die Prognose nutzbar zu machen. Ähnliches gilt für die Daten zu Ankünften von LKWs und Zügen im Hafen. Hier liegen die Daten erst in nachgelagerten Auswertungen in digitaler Form vor, was sie für die Prognosemodelle im Echtbetrieb nicht nutzbar macht.

#### **2.2.4.4 Datenvorverarbeitung**

Damit mit der Vorverarbeitung der Daten begonnen werden konnte, musste zunächst die Datenerfassung abgeschlossen werden. In einem mehrtägigen Workshop im Oktober 2021 wurde gemeinsam mit den Partnern die Bereinigung der Datenerfassung abgeschlossen. Dies umfasste die Überprüfung der erfassten Werte jeder einzelnen Zählstelle und die technische Anpassung, sollten Unstimmigkeiten auffallen. Anschließend konnte die Datenerfassung beginnen. Um möglichst schnell mit der Entwicklung der Modelle beginnen zu können, wurden über drei Monate Daten erfasst, welche anschließend die Grundlage für die Modellentwicklung und das Training bildeten. Um die Eingabe- und Ausgabedaten in das gewünschte Format zu übertragen ist eine Datenvorverarbeitung notwendig. Da die Eingabedaten bei NPorts im 15 Sekunden Abstand, und bei J. Müller nur bei einer eingetretenen Änderung vorliegen, müssen diese Werte auf die 15-Minuten Intervalle aggregiert werden. Dazu werden zunächst fehlende Werte der 15-Sekunden Intervalle aufgefüllt. Bei J. Müller kann erwartet werden, dass Werte fehlen, wenn keine Änderung auftritt, aber auch bei NPorts kann es durch technische Fehler vorkommen, dass vereinzelt Werte fehlen. Diese werden im ersten Schritt durch den letzten bekannten Wert ersetzt, sodass für jedes 15-Sekunden Intervall im betrachteten Zeitraum ein Leistungswert vorliegt. Aus diesen Werten für 15-Sekunden Intervalle werden dann die Werte der 15-Minuten Intervalle aggregiert, indem der Mittelwert gebildet wird. Da die Modelle immer vier Werte als Eingabe erhalten und vier Werte als Ausgabe liefern, müssen diese Paare aus den Daten des betrachteten Zeitraums gebildet werden. Diese Paare werden anschließend als Datenpunkte abgelegt und für das Training der Modelle verwendet.

#### **2.2.4.5 Prognosemodelle**

Grundsätzlich sind für das beschriebene Problem alle Verfahren des supervised learning geeignet. Das können statistische Verfahren wie lineare Regression, als auch KI-Methodiken wie künstliche neuronale Netze sein. Da es sich bei den zur Verfügung stehenden Daten um Zeitreihendaten handelt, bieten sich aber besonders rekurrente neuronale Netze als Prognosemodelle an. Diese zeichnen sich dadurch aus, eine Art Gedächtnis zu haben, durch das vorhergehende Zeitschritte bei der Prognose berücksichtigt werden können. Das ist besonders für Zeitreihen geeignet, da es so ermöglicht wird die Verläufe der Zeitreihen in die Prognose einfließen zu lassen. Eine spezielle Form von rekurrenten neuronalen Netzen sind Long Short Term Memory (LSTM) Netze. Sie zeichnen sich durch ihre Fähigkeit aus, über einen langen Zeitraum ein Kurzzeitgedächtnis zu speichern und somit sehr stabil in der Vorhersage von Zeitreihendaten zu sein.<sup>4</sup> Um diese Vorteile für die Prognose der Lastspitzen zu nutzen, basieren die entwickelten Modelle auf LSTM Netzen.

Da die Prognosemodelle nur auf den Lastverläufen basieren, ist eine Übertragbarkeit auf andere Standorte grundsätzlich möglich. Es ist allerdings eine ähnliche Art der Datenerfassung notwendig, auf deren Grundlage dann die Modelle für den neuen Standort trainiert werden können. Also muss zunächst eine Datenerfassung aufgebaut werden, über einen Zeitraum von mehreren Monaten Daten erfasst werden und diese dann genutzt werden um neue Prognosemodelle zu trainieren.

---

<sup>4</sup> Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Diese können dann in das Gesamtsystem, welches auf den neuen Standort übertragen wurde, integriert werden

#### **2.2.4.6 Prognosemodelle für Handlungsempfehlungen**

Relativ spät im Projekt wurde ein weiteres Prognosemodell konzipiert. Die Idee war, die zu dem Zeitpunkt zur Verfügung stehenden Daten zu Schiffspositionen im Hafen zu nutzen, um eine Prognose über die abgerufene Leistung der einzelnen Verbraucher zum Zeitpunkt der Handlungsempfehlung zu geben. Im Test hat sich allerdings ergeben, dass die Prognose einzelner Verbraucher dazu führen kann, dass Handlungsempfehlungen gegeben werden, Verbraucher einzuschalten, die zu dem Zeitpunkt noch gar nicht eingeschaltet sind. Um eine Verwirrung der Anwender zu vermeiden, wurde diese Prognose deshalb aus dem System wieder entfernt. Da die Modelle aber komplett konzipiert und entwickelt wurden, werden sie hier trotzdem beschrieben.

Da die Handlungsempfehlungen für einen Zeitpunkt in der Zukunft gegeben werden, sollte ein Prognosemodell entwickelt werden, um die abgerufene Leistung der einzelnen Verbraucher unter der Einspeisung, für die eine Handlungsempfehlung generiert wird, zu prognostizieren. Analog zum oben beschriebenen Prognosemodell werden die Leistungswerte der letzten Stunde als Eingabewerte verwendet. Hier allerdings in einminütiger Auflösung. Um die Prozesse im Hafen besser abbilden zu können, werden die erst spät im Projekt verfügbar gewordenen Daten der installierten AIS Antenne zusätzlich zu den Lastverläufen eingesetzt. Die Auswertung und Verbindung der AIS-Daten mit den Leistungsdaten ist sehr Zeitaufwändig und Rechenintensiv, weshalb diese auf einen Server ausgelagert werden mussten und die Datenvorverarbeitung aufwändig neu designt werden musste. Auch für diese Modelle wurden LSTM Modelle eingesetzt, da die gleichen Anforderungen wie bei den anderen Prognosemodellen gelten. Aufgrund der erhöhten Datenmenge war mit deutlich erhöhten Trainingszeiten zu rechnen, weshalb auch das Training der Modelle auf spezialisierte KI-Hardware ausgelagert wurde. Die Ausgabe der Prognosemodelle sind die prognostizierten Leistungswerte zum Zeitpunkt der Handlungsempfehlung. Auch die Einbindung in das Gesamtsystem wurde im Projekt durchgeführt, aber aufgrund der oben beschriebenen Bedenken aus Nutzersicht wurden diese Modelle lediglich in einer kurzen Testphase im System eingesetzt.

#### **2.2.5 (TAP 6.1) Konzeptionierung dashPORT**

Im Teilarbeitspaket 6.1 wurde vom OFFIS die Konzeptionierung des dashPORT-Systems durchgeführt. Das beinhaltet die Erhebung und Zusammenführung der Anforderung. Basierend auf den Anforderungen wurden dann Software-Architektur und Benutzeroberfläche entwickelt.

##### **2.2.5.1 Anforderungserhebung**

Die Anforderungen an das dashPORT-System wurden von verschiedenen Quellen im Projekt abgeleitet. In erster Linie wurde dafür die Gesamtvorhabensbeschreibung herangezogen. Eine weitere Quelle stellten die Workshops und Ergebnisse vom Fraunhofer CML zur Erhebung der Prozesse im Hafen dar. Über die gesamte Projektlaufzeit wurden zwei große UI-Workshops mit allen Projektpartnern durchgeführt, einer zu Beginn, um initiale Anforderungen zu erheben und einer zum Ende, um das Design der Handlungsempfehlungen zu finalisieren. Außerdem wurde der Designfortschritt in den monatlichen Treffen diskutiert und Feedback sofort berücksichtigt. Basierend darauf wurden folgende Anforderungen abgeleitet.

###### *A1.1 Zusammenführung von getrennt erhobenen Daten*

Die von J. Müller und NPorts getrennt erhobenen Messdaten sollten von dashPORT zusammengeführt und in sinnvoller Weise dargestellt werden.

#### *A1.2 Darstellung historischer, aktueller und prognostizierter Verbräuche*

Die Benutzeroberfläche sollte in der Lage sein alle gemessenen und prognostizierten Verbrauchsdaten darzustellen.

#### *A1.3 Darstellung außergewöhnlicher Events*

Die Oberfläche sollte auf außergewöhnliche Events, wie prognostizierte Lastspitzen, hinweisen und den Nutzer in geeigneter Weise darüber informieren.

#### *A1.4 Darstellung von Handlungsempfehlungen*

Die Oberfläche sollte vom System generierte Handlungsempfehlungen darstellen.

#### *A1.5 Prognose je Einspeisung*

Das System soll je Einspeisung die nächste Stunde in 15-Minuten-Werten prognostizieren.

#### *A1.6 Erkennen von Lastspitzen*

Das System soll auf Basis der Prognose die nächste Lastspitze erkennen.

#### *A1.7 Ermitteln von Handlungsempfehlungen*

Das System soll, wenn eine Lastspitze bevorsteht, Handlungsempfehlungen ermitteln und dem Benutzer anzeigen.

#### *A1.8 Übertragbarkeit auf andere Standorte*

Das System soll auf andere Standorte von NPorts übertragbar sein.

#### *A1.9 Einsatz von Datenstrommanagementsystem*

Das System soll die Datenverarbeitung über ein Datenstrommanagementsystem realisieren

#### *A1.10 Einsatz von Machine Learning Methoden*

Die Prognose wird mit Hilfe von Machine Learning Methoden umgesetzt werden.

#### *A1.11 Datenübertragung mittels IEC 60870-5-104*

Die Messdaten von NPorts werden durch IEC 60870-5-104 übermittelt

#### *A1.12 Datenübertragung mittels OPC-Server*

Die Messdaten von J. Müller müssen über einen OPC-Server abgerufen werden.

Aus den durchgeführten UI-Workshops und den regelmäßigen Feedbackrunden ergaben sich die folgenden weiteren Anforderungen:

#### *A2.1 Einfache Darstellung*

Die Oberfläche sollte möglichst intuitiv und leichtverständlich sein, da dashPORT über eine heterogene Nutzerbasis verfügt.

#### *A2.2 Reduzierte Informationsdichte*

Die Oberfläche sollte möglichst nur die Informationen darstellen, die aktuell benötigt werden, um Leitstandmitarbeiter zu entlasten, die viele Monitoring-Aufgaben durchzuführen haben.

#### *A2.3 Notwendige Informationen "Auf einen Blick"*

Die Oberfläche sollte möglichst alle nötigen Informationen auf einer Seite darstellen, ohne dass unnötig viel gescrollt oder in Untermenüs nach Informationen gesucht werden muss.

#### *A2.4 Verwaltung von Handlungsempfehlungen ermöglichen*

Die Oberfläche soll es den Nutzern erlauben Handlungsempfehlungen anzunehmen und zu archivieren.

#### *A2.5 Separate Prognose nach Einspeisungen*

Da dashPORT mit dem Ziel entwickelt wurde Lastspitzen zu verhindern, um Kosten zu senken, sollte die Prognose nicht aggregiert über alle Einspeisungen erfolgen, sondern separat für jede Einspeisung. Die einzelnen Einspeisungen sind Abrechnungsrelevant.

#### *A2.6 Zusätzliche Funktion "Verbrauchsabschätzung"*

Um Verbrauchsintensive Prozesse verstehen und planen zu können, wurde eine zusätzliche Funktion gewünscht, die es ermöglicht den Verbrauch einzelner Verbraucher und ihren Einfluss auf den Gesamtverbrauch im Hafen zu visualisieren.

#### *A2.7 Historische Betrachtung von Lastspitzen*

Neben der Betrachtung von historischen Messdaten sollte als zusätzliche Funktion ebenfalls die historische Betrachtung von Lastspitzen ermöglicht werden.

#### *A2.8 Handlungsempfehlungen als E-Mail-Analogie*

Die Benutzeroberfläche sollte Handlungsempfehlungen analog zu modernen E-Mail-Visualisierungen darstellen, um einen intuitiven Umgang zu ermöglichen.

#### *A2.9 Handlungsempfehlungen individualisieren*

Handlungsempfehlungen beinhalten Listen von Verbrauchern, die ausgeschaltet werden können. Die Oberfläche sollte es ermöglichen die unterschiedlichen Vorschläge aus den Listen zu kombinieren, um den Nutzern mehr Flexibilität zu ermöglichen.

### **2.2.5.2 Technische Beschreibung des Gesamtsystems**

Abgeleitet von den vorher beschriebenen Rahmenbedingungen und Anforderungen wurde die nachfolgenden Systemarchitektur entwickelt. Bei dem Entwurf der Architektur wurde sich an bekannte Big-Data- und IoT-Architekturen orientiert. Dazu gehören die Lambda- und Kappa-Architektur. Beide Architekturen sind darauf ausgelegt einen kontinuierlichen Strom von Daten zu verarbeiten und die Daten für weitere Anwendungen bereitzustellen. Dieser Ansatz eignet sich für die Problemstellungen in dashPORT. Als kontinuierlichen Datenstrom erhält das System Daten von der Messpunkten im Feld. Aus dem Datenstrom muss das System anschließend Prognosen und Handlungsempfehlungen generieren und sie sowohl für historische Analysen abspeichern als auch dem Dashboard bereitstellen.

Die Lambda-Architektur besteht aus vier Komponenten. Das Data Ingestion Layer erhält Daten aus den verschiedenen Datenquellen, bereitet die Daten auf und leitet sie weiter an das Speed-Layer und Batch-Layer. Im Speed-Layer werden Daten in Echtzeit durch Datastream-Processing verarbeitet. Im Batch-Layer werden die Daten erst gesammelt und in einem Batch verarbeitet. Beide Layer stellen anschließend im Serving-Layer die Daten anderen Diensten bereit. Dieser

kann beispielsweise durch eine Datenbank oder einem Message-Broker realisiert werden.

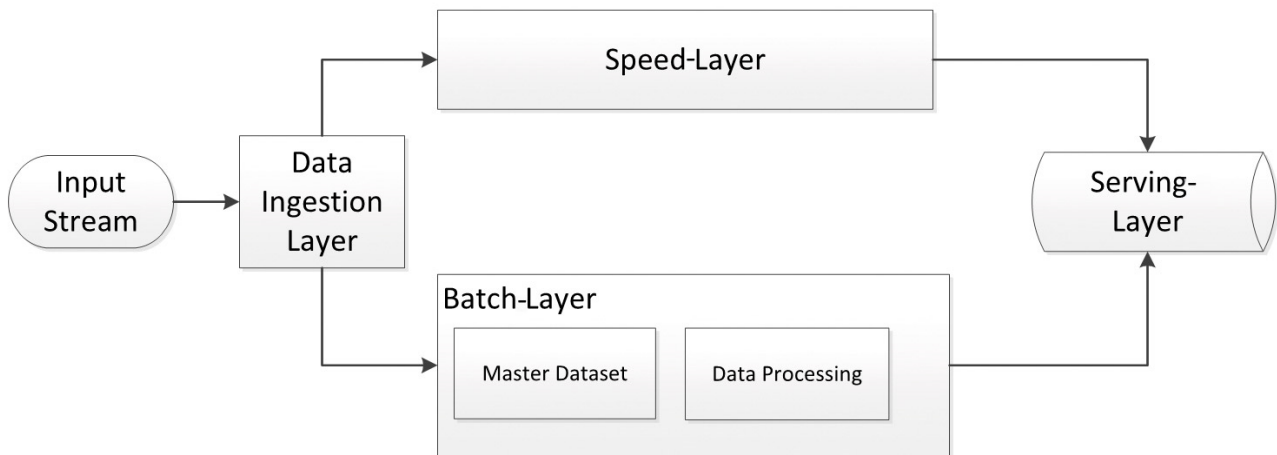


Abbildung 4: Lambda-Architektur<sup>5</sup>

Die Kappa-Architektur ist der Lambda-Architektur sehr ähnlich. Sie unterscheiden sich darin, dass der Batch-Layer nicht vorhanden ist und Daten aus den Quellen neben dem Speed-Layer in einem Master Dataset gespeichert werden. Es werden jedoch nicht bereits verarbeitete Daten abgespeichert. Das bedeutet, wenn auf historische Ergebnisse zugegriffen werden soll, müssen die Daten nochmal vom Speed-Layer verarbeitet werden.

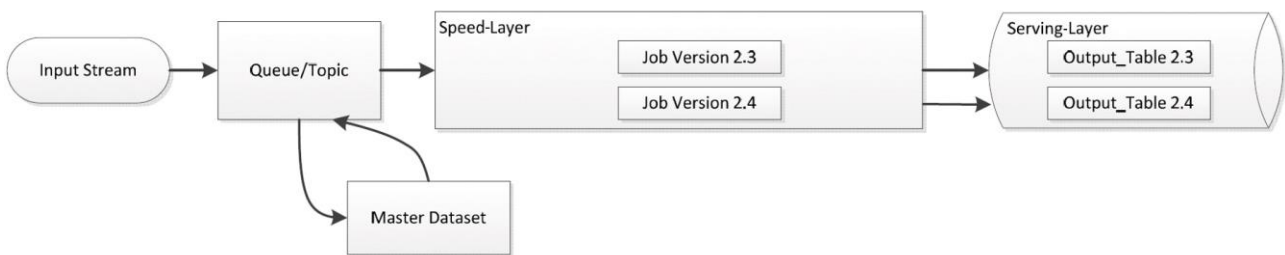


Abbildung 5: Kappa-Architektur<sup>6</sup>

In Anbetracht der Anforderungen, dass häufig auf historische Daten zugegriffen wird, wurde sich für eine Architektur angelehnt an die Lambda-Architektur entschieden. Sowohl die Data-Ingestion als auch das Speed-Layer wird durch Apache Flink realisiert. Das Batch-Layer wird durch InfluxDB realisiert. Über das Flux-Interface der Datenbank können für das Projekt ausreichende Anfragen über die gesamte Datenbasis durchgeführt werden. Das Serving Layer wird innerhalb der Dashboard-Architektur realisiert. Die Kommunikation zwischen den Schichten wird von RabbitMQ

<sup>5</sup> <https://entwickler.de/software-architektur/lambda-vs-kappa>

<sup>6</sup> <https://entwickler.de/software-architektur/lambda-vs-kappa>



als Message-Broker übernommen.

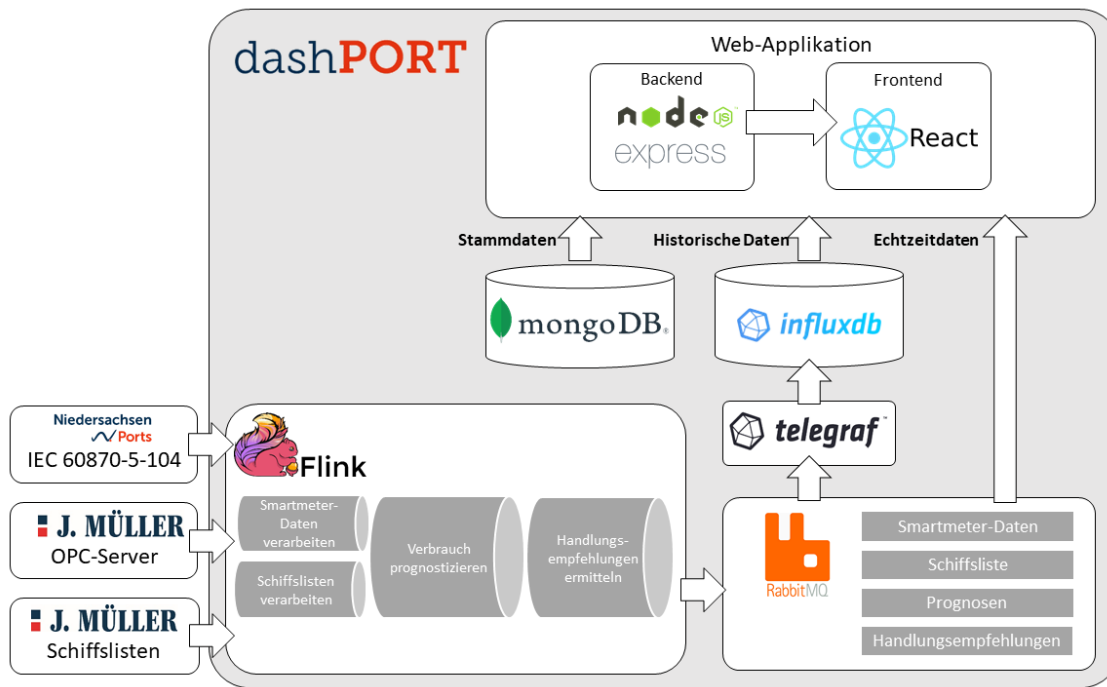


Abbildung 6: Softwarearchitektur der digitalen Leitwarte

## Technologieentscheidungen

Als Message-Broker wurde sich für RabbitMQ gegenüber Apache Kafka entschieden. Kafka ist zwar De-facto-Standard in Big-Data-Systemen, jedoch ergaben die Workshops das bereits MQTT in verschiedenen Bereichen bei NPorts im Einsatz ist. Um daher eine bessere Übertragbarkeit zu gewährleisten wurde sich für RabbitMQ entschieden.

Da es sich bei den Messdaten um Zeitreihen handelt wird eine Zeitreihendatenbank eingesetzt. In der Datenbank soll über die gesamte Projektlaufzeit Daten gespeichert werden, daher ist es wichtig, dass sie bereits am Markt etabliert und stabil ist. Aus diesem Grund wurde sich dafür entschieden InfluxDB zu verwenden. Im Zusammenhang mit InfluxDB wird auch Telegraf eingesetzt, um ohne weiteren Implementierungsaufwand Daten von RabbitMQ in der Datenbank speichern zu können.

Bei der Technologieentscheidung zum Datenstrommanagementsystem wurde darauf Rücksicht genommen, dass im Machine-Learning-Umfeld hauptsächlich mit Python entwickelt wird. Weshalb ein Python-API vorhanden sein sollte. Daher wird Apache Flink verwendet.

## Datenfluss

Sowohl Messwerte von allen Smartmetern also auch Daten anderen Datenquellen, die regelmäßig aktualisiert werden, werden von Flink abrufen oder zu Flink geschickt. Dafür wird für jede Datenquelle ein eigener Adapter verwendet.

In einer nachgelagerten Pipeline werden dann alle Daten aufbereitet und verarbeitet. Nach jedem Pipeline-Schritt werden, die Daten an RabbitMQ gesendet und dort sowohl vom nächsten Pipeline-Schritt abgerufen, als auch weiter über Telegraf in der InfluxDB gespeichert.

In Flink wird neben dem Aufbereiten der Daten auch die Prognose über Machine-Learning-Modelle durchgeführt und im Falle einer Lastspitze auch Handlungsempfehlungen generiert.

Messwerte, Prognosen und Handlungsempfehlungen können vom Dashboard entweder direkt vom RabbitMQ abgerufen werden, um Echtzeitdaten anzuzeigen oder es können historische Daten aus der Datenbank abgerufen werden. Zusätzlich stehen dem Dashboard über die MongoDB Stammdaten zu jedem Zähler zur Verfügung.

## **Übertragbarkeit**

Eine wichtige Anforderung von NPorts ist die Übertragbarkeit des Systems auf andere Standorte des Unternehmens. Im Kontext der Software-Architektur wird nur auf die technische Übertragbarkeit eingegangen.

Um die Übertragbarkeit des Systems herauszuarbeiten, wurden Gespräche mit Mitarbeitern von anderen NPorts-Standorten durchgeführt. Das Ergebnis war, dass häufig schon ein MQTT-basierter Message-Broker vorhanden ist. Auch eine Bereitstellung von Messwerten über den Message-Broker stellt kein Problem dar. Daher wurde eine alternative Architektur vorgeschlagen. Das dashPORT-System besteht dabei nur aus der Datenbank, dem Dashboard und einer Komponente zum generieren der Prognose und Handlungsempfehlungen. Dadurch ist das System besser integriert und es müssen nicht zusätzliche Adapter zur Datenerfassung implementiert werden. Auf Apache Flink wird in dem Fall auch verzichtet, da es nur noch die Prognose und Handlungsempfehlungen übernehmen würde. Das reduziert weiter den Technologie-Stack und erleichtert die Wartbarkeit des Systems.

### **2.2.5.3 Entwurf der grafischen Benutzeroberfläche**

Die grafische Benutzeroberfläche (en.: User Interface (UI)) für dashPORT wurde nach der User-Centered Design-Methodik<sup>7</sup> entworfen. Diese iterative Methodik stellt die potenzielle Nutzerbasis in den Vordergrund und legt den Fokus auf eine hohe Benutzerfreundlichkeit. Zunächst erfolgt eine Analyse der Nutzerbasis, indem Ziele und Anforderungen definiert werden. Aus diesen Informationen wird ein Design konzeptioniert und umgesetzt. Das fertige Design wird anschließend mit der Nutzerbasis evaluiert. Feedback aus der Evaluation dient als Grundlage für die nächste Iteration.

## **Entwurfsprozess**

Über die gesamte Projektlaufzeit sind mehrere Oberflächenkonzepte entstanden. Im Folgenden wird das Endergebnis dargestellt.

---

<sup>7</sup> Abras et al. (2004) User-Centered Design. In: Bainbridge, W., Ed., Encyclopedia of Human-Computer Interaction, Sage Publications, Thousand Oaks, CA, 445-456

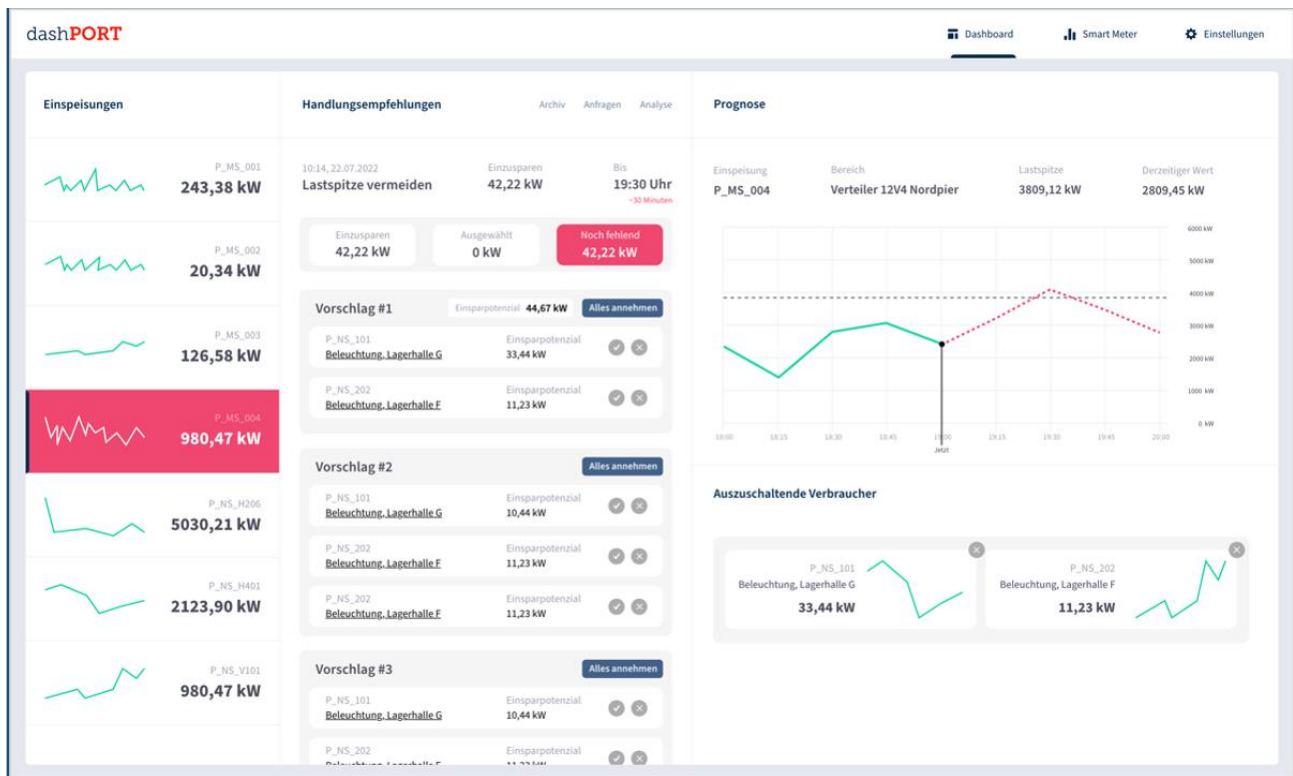


Abbildung 7: DashPORT Entwurf "Startseite"

Das Oberflächenkonzept von dashPORT sieht zwei Hauptansichten vor. Die "Startseite" (vgl. Abbildung 7) als zentrales Element des Systems, mit einem Überblick über die Energieflüsse im Hafen, besonderen Events und Einspeisungen. Und die "Smartmeter"-Seite mit Zugriff auf alle historischen Verbrauchsdaten.

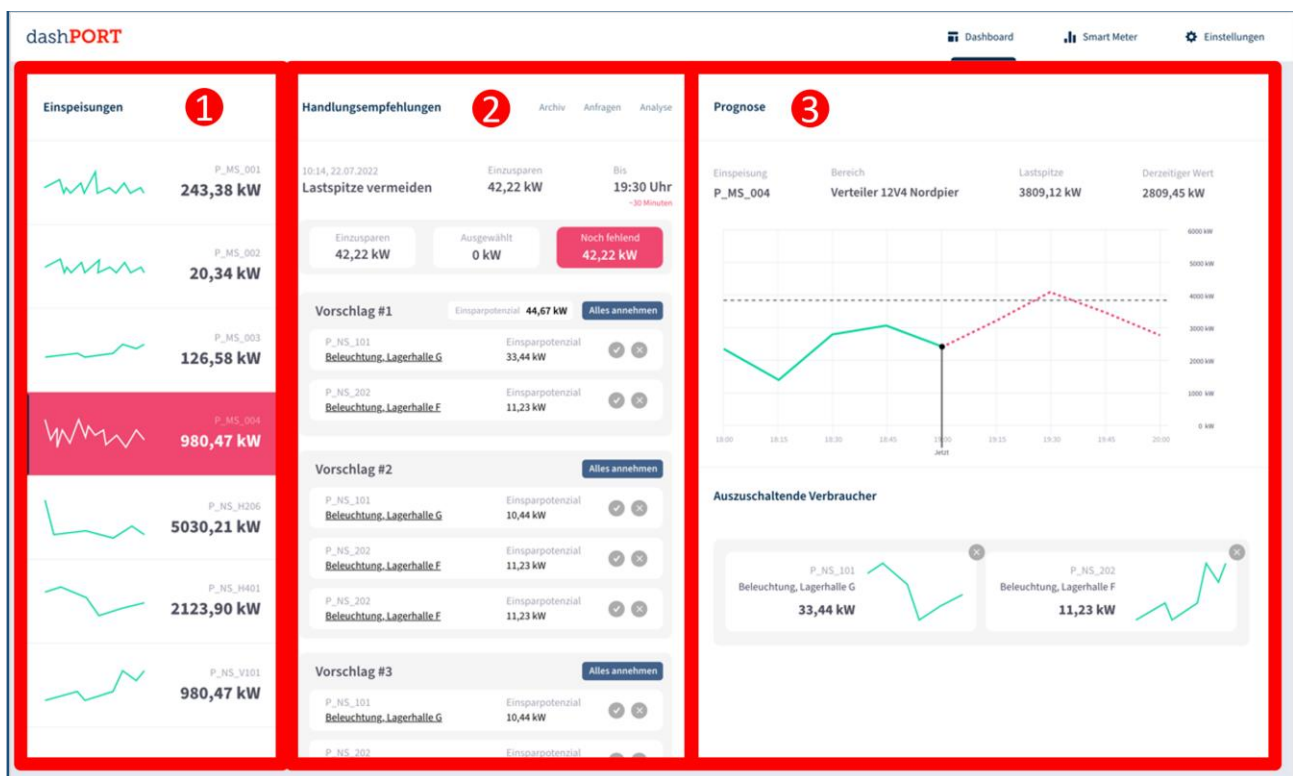


Abbildung 8: DashPORT Entwurf "Startseite" – Überblick über Bereiche

Die Startseite ist in drei Bereiche unterteilt: Einspeisungen, Handlungsempfehlungen und Prognose.

Im Bereich **Einspeisungen** (vgl. Abbildung 8-1) findet sich eine Liste aller Einspeisungen des jeweiligen Projektpartners. Diese Liste fungiert als zentrales Steuerelement der Seite. Durch Anklicken einer Einspeisung werden die zugehörigen Handlungsempfehlungen und Prognosewerte sichtbar. Jede Einspeisung verfügt über ein kleines Liniendiagramm auf der linken Seite, einen Bezeichner auf der rechten Seite und den aktuellen Messwert in kW. Jede Einspeisung verfügt über zwei Zustände: weiß und rot. Weiß bedeutet, dass keine Handlungsempfehlungen vorliegen, rot, dass mindestens eine Handlungsempfehlung eingegangen ist und eine Aktion seitens des Nutzers erforderlich ist.

Im Bereich **Handlungsempfehlungen** (vgl. Abbildung 8-2) findet sich eine Liste aller eingegangenen Handlungsempfehlungen zu der ausgewählten Einspeisung. Jede Handlungsempfehlung verfügt über einen Zeitstempel, über einen Wert der angibt wie viel kW einzusparen sind und bis wann. Darunter befindet sich ein Hilfsindikator, der visualisiert, ob die ausgewählten Verbraucher genügend kW einsparen würden, um die prognostizierte Lastspitze zu verhindern. Unter dem Indikator befindet sich eine Liste mit Vorschlägen und in jedem Vorschlag eine Liste mit Verbrauchern. Eine Besonderheit der Oberfläche ist, dass die Verbraucher aus den verschiedenen Vorschlagslisten miteinander kombiniert werden können, sollte ein Vorschlag aus Prozesssicht gerade nicht umgesetzt werden können.

Wurden Verbraucher oder ein gesamter Vorschlag angenommen, werden die ausgewählten Verbraucher unter dem Bereich "Angenommen" zusammengefasst und die übrigen Vorschläge eingeklappt, um die Informationsdichte zu reduzieren (vgl. Abbildung 9).

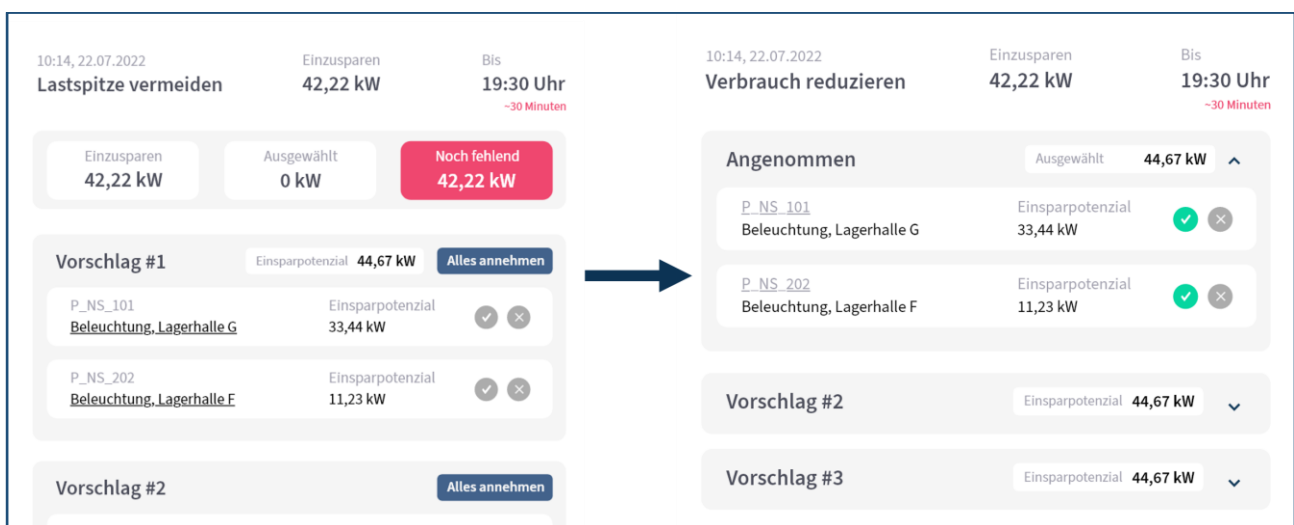


Abbildung 9: Anpassung der Benutzeroberfläche an angenommene Handlungsempfehlungen

Im Kopfbereich des Bereichs befinden sich drei weitere Auswahlmöglichkeiten: Archiv, Anfragen, Analyse.

Über das **Archiv** können archivierte Handlungsempfehlungen eingesehen werden.

Über die **Anfragen** können Verbrauchsabschätzungen auf den vorhandenen Daten durchgeführt werden (vgl. Abbildung 10). Im oberen Suchfeld wird der Bezeichner eines Smartmeters oder einer Einspeisung angegeben. Durch Anklicken des Verbrauchers erhält der Nutzer einen Überblick über den durchschnittlichen und den maximalen Verbrauch seit Anfang des Jahres. Wird

“Berechnen” angeklickt, aktualisiert sich das Prognosediagramm auf der rechten Seite und zeichnet zwei weitere Kurven für den durchschnittlichen und maximalen Verbrauch ein.

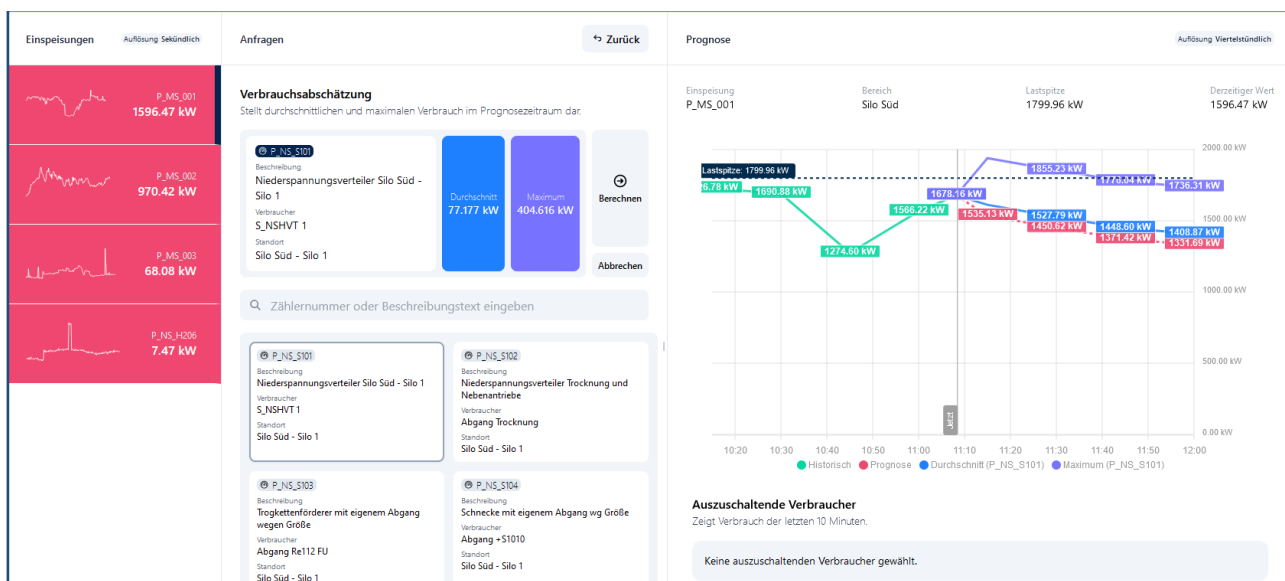


Abbildung 10: dashPORT Benutzeroberfläche "Verbrauchsabschätzung"

Über die **Analyse** können die Bestandsdaten in Bezug auf historische Lastspitzenüberschreitungen analysiert werden (vgl. Abbildung 11). In der mittleren Spalte befindet sich eine Liste aller gemessenen Lastspitzenüberschreitungen seit Beginn des Jahres. Die Höhe der Überschreitung wird durch einen Wert am rechten Rand und einer farblichen Markierung visualisiert. Je dunkler die Fläche um den Wert, desto höher die Überschreitung. Durch Anklicken einer Lastspitze erscheint auf der rechten Seite eine Analyse. Ein Sankey-Diagramm visualisiert die zehn größten Verbraucher, darunter befindet sich eine Liste, in der alle Verbraucher absteigend nach Höhe ihres Verbrauchs aufgelistet sind.

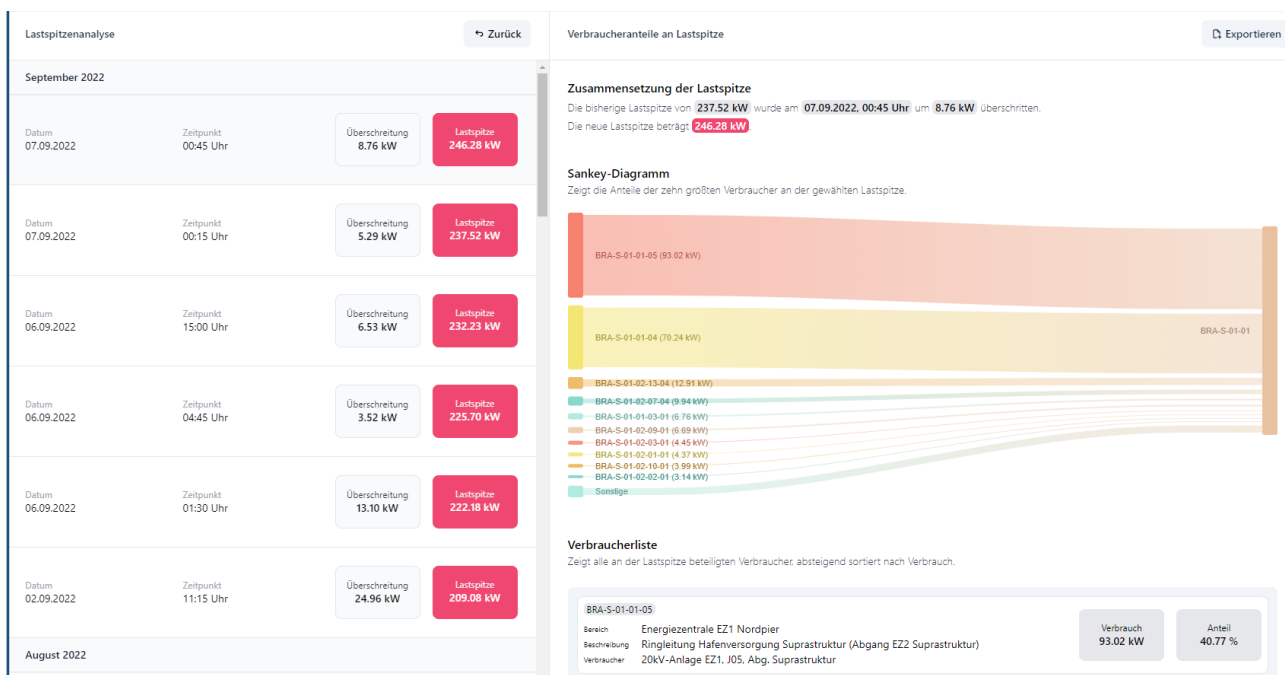


Abbildung 11: dashPORT Benutzeroberfläche „Analyse“

Im Bereich **Prognose** befindet sich die Prognosesicht (vgl. Abbildung 12). Das Diagramm stellt in grün den historischen Verbrauch der letzten Stunde dar und in rot den prognostizierten Verbrauch

der nächsten Stunde. Eine blaue gestrichelte Linie zeigt die Höhe der letzten Lastspitze, die unterschritten werden soll, an. Unter dem Diagramm findet sich eine Liste aller auszuschaltenden Verbraucher, die in den Handlungsempfehlungen ausgewählt wurden.

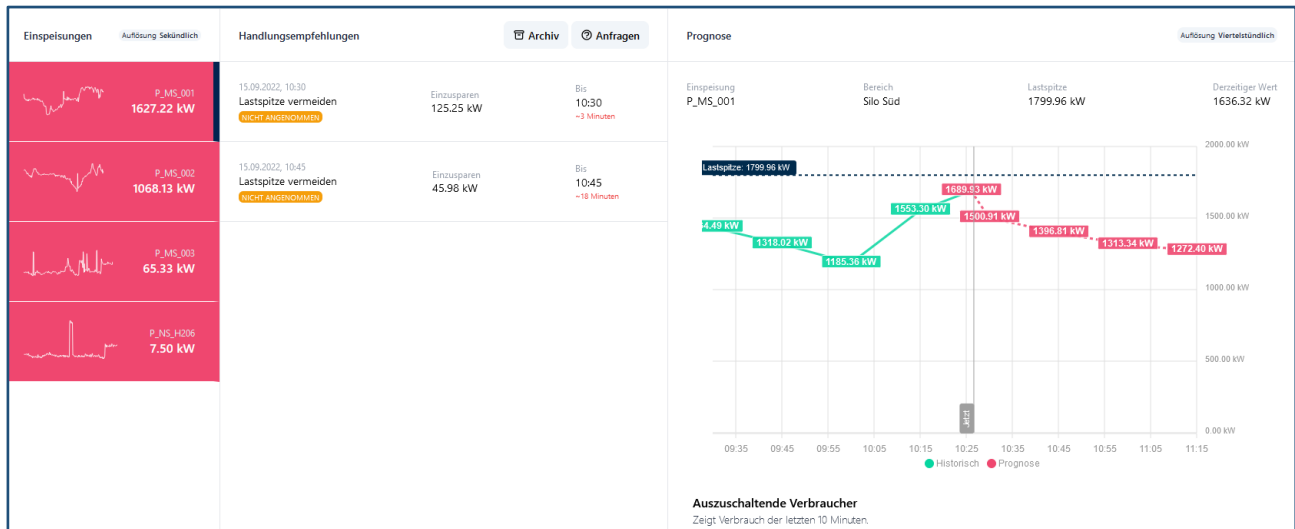


Abbildung 12: dashPORT Benutzeroberfläche "Prognose"

## Mobile Ansicht

Für Hafenmitarbeiter, die im Feld agieren und Handlungsempfehlungen umsetzen sollen, wurde eine angepasste mobile Ansicht entwickelt (vgl. Abbildung 13).

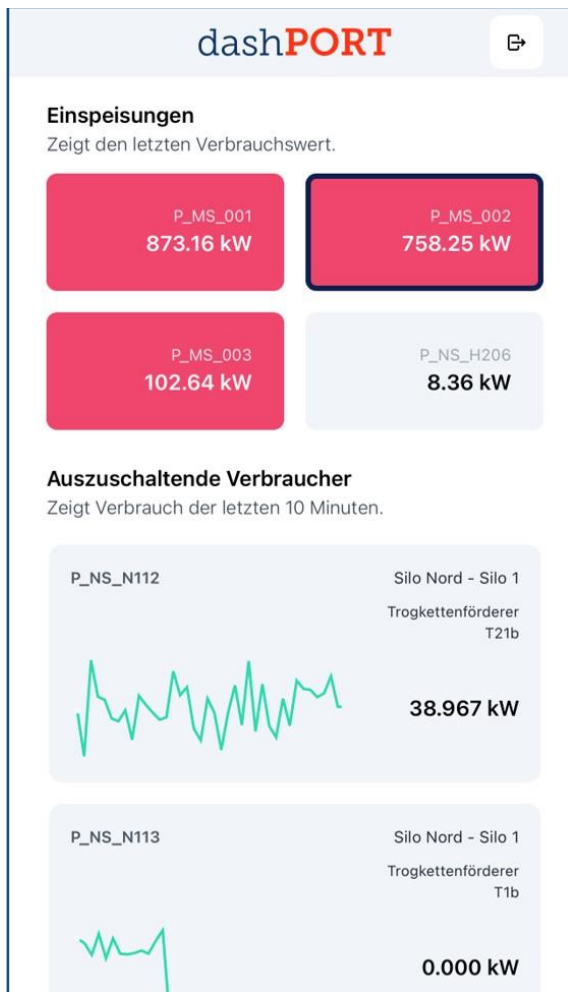


Abbildung 13: Mobile Ansicht der Benutzeroberfläche

Im oberen Bereich befindet sich eine Liste mit Einspeisungen, die ebenfalls über die unter "Einspeisungen" beschriebenen Funktionen verfügt. Darunter befindet sich die Liste mit auszuschaltenden Verbrauchern. Handlungsempfehlungen oder weitere Funktionen können nicht verwendet werden.

Neben dem Dashboard gibt es in dashPORT auch noch eine Detailansicht, welche zur Darstellung historischer Verbräuche dient (vgl. Abbildung 14).



Abbildung 14: dashPORT Benutzeroberfläche "Detailansicht"

#### **2.2.5.4 Entwurf des Rollenkonzepts**

Ziel des Rollenkonzepts war es verschiedene Sichten auf den Datenbestand und Einschränkungen bei der Bedienung der Benutzeroberfläche zu realisieren. Als Ergebnis des ersten UI-Workshops wurde definiert, dass es zwei Nutzerrollen geben soll: Bearbeiter und Mitarbeiter. Der Bearbeiter soll Handlungsempfehlungen annehmen und archivieren können, der Mitarbeiter soll sie lediglich sehen können.

Im Projektverlauf hat sich gezeigt, dass lediglich Personen die im Feld agieren und Handlungsempfehlungen umsetzen über Mitarbeiter-Rechte verfügen sollen, während alle anderen Nutzer Bearbeiter-Rechte benötigen. Aus diesem Grund wurde das Rollenkonzept durch zwei unterschiedliche Oberflächen, einem Client-Dashboard und einer mobilen Ansicht, realisiert.

Die Administration der Zugriffe auf den Datenbestand findet über InfluxDB selbst statt. Die Administration der Nutzer erfolgt über MongoDB.

#### **2.2.6 (TAP 6.2) Implementierung dashPORT**

Das Teilarbeitspaket 6.2 beschäftigt sich mit der Implementierung des dashPORT-Systems nach dem in Teilarbeitspaket 6.1 entwickelten Konzept. Es wird beschrieben wie die ausgewählten Technologien eingesetzt werden und Software-Komponenten implementiert wurden.

#### **2.2.3.2 Datenbanksystem (InfluxDB, Telegraf)**

Innerhalb der InfluxDB gibt es je ein Bucket für J.Müller und NPorts um die Daten getrennt voneinander zu speichern und ggfs. Zugriffe einzuschränken. Innerhalb der jeweiligen Buckets wird die identische Datenstruktur für beide Unternehmen verwendet.

Es gibt zwei Arten von Zeitserien, die das System verarbeiten können muss. Einerseits müssen für jeden Messpunkt die unverarbeiteten Messwerte abgespeichert werden. Andererseits werden alle prognostizierten Werte je Einspeisung abgespeichert. Bei den Messwerten handelt es sich um 15 Sekunden kWh- und kW-Werte. Auf eine Speicherung in anderen Auflösungen wird verzichtet. Falls eine andere Auflösung benötigt wird, kann diese über die Anfragesprache Flux berechnet werden.

Bei den prognostizierten Werte handelt es sich um 15 Minuten kW-Werte. Die Datenmenge ist in dem Fall wesentlich geringer auch weil Prognosen nur für die Einspeisungen und nicht alle Messpunkte durchgeführt werden.

Um Daten zu speichern wird Telegraf verwendet. Hierfür wurde Telegraf so konfiguriert, dass jedes RabbitMQ-Topic in den jeweiligen Buckets in InfluxDB gespeichert wird. Das Dashboard und Flink haben lesenden Zugriff auf die Datenbank. Die Datenstruktur innerhalb von InfluxDB ist in Abbildung 15 dargestellt. Da es sich um eine Zeitreihendatenbank handelt, wird der Zeitstempel nicht mit aufgeführt. Falls kein Zeitstempel mitgegeben wird, setzt die Datenbank selber einen.

Für Messwerte von NPorts wird der mitgesendete Zeitstempel verwendet. Für Messwerte von J. Müller wird der Zeitstempel erfasst, sobald die Messwerte vom OPC-Server abgerufen werden.



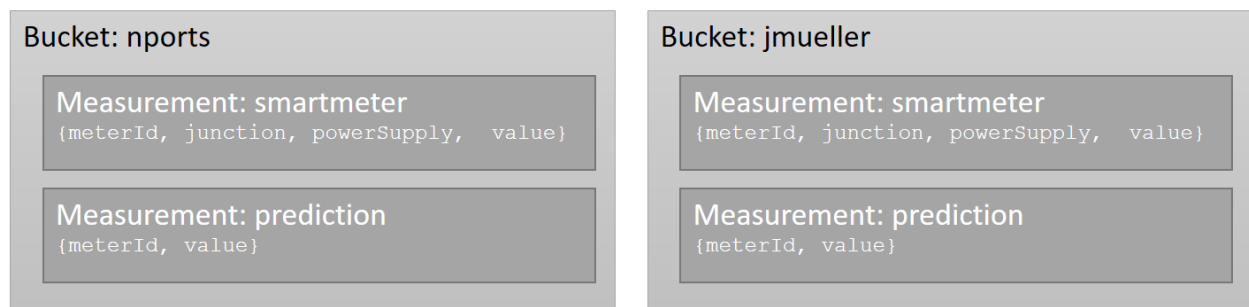


Abbildung 15: Datenstruktur InfluxDB

### 2.2.3.1 Datenstrommanagementsystem (Flink, RabbitMQ)

Auf dem Datenstrommanagementsystem Apache Flink werden die Messdaten aus dem Hafen in Echtzeit verarbeitet. Dabei werden die Verarbeitungsschritte in Flink-Jobs gegliedert. Nach jedem Job werden die Daten an RabbitMQ gesendet und dort vom nächsten Job abgeholt und von Telegraf in der Datenbank gespeichert.

Ursprünglich war geplant, die Flink-Jobs für die Prognose und den Handlungsempfehlungen in Python über die Flink-Python-API zu implementieren. Während der Implementierung stellte sich jedoch heraus, dass die Python-API nicht den gleichen Funktionsumfang wie die Java-API hat. Beide Komponenten über die Java-API zu implementieren, stellte sich als ungeeignet heraus. Aus diesem Grund wurden die Funktionen in einer eigenen Komponente ausgelagert und mittels Python implementiert. Daraus resultiert die in Abbildung 16 zu sehende Gliederung der Flink-Jobs.

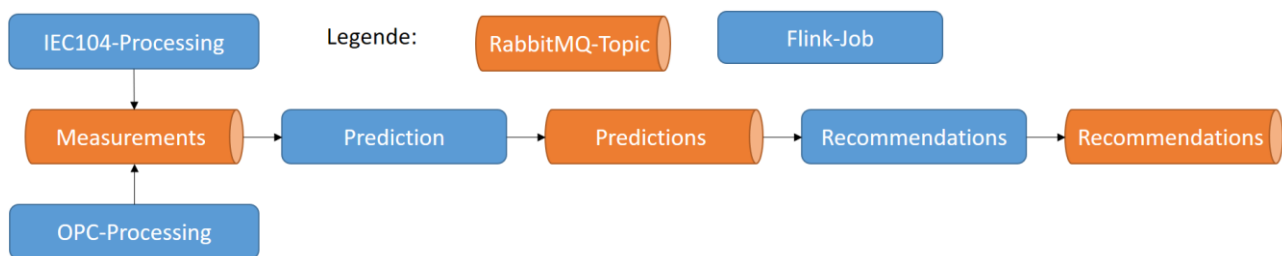


Abbildung 16: Flink-Dataflow

Im Folgenden wird auf die einzelnen Verarbeitungsschritte der Jobs eingegangen

#### IEC-104-Adapter

Der IEC104-Adapter nimmt als Client eine TCP-Verbindung nach dem IEC 60870-5-104-Protokoll zum Server bei NPorts auf. Anschließend sendet der IEC-104-Server alle 15 Sekunden die Messwerte aller Messpunkte.

Die Implementierung mithilfe der Java-Flink-API statt. Zusätzlich wird die Java-Bibliothek OJ104<sup>8</sup> zur Umsetzung des IEC104-Protokolls verwendet.

Sobald Messwerte empfangen werden, werden sie wie im Anfrageplan dargestellt verarbeitet. Als erstes werden die Nachrichten nach den IEC-104-Nachrichtentypen 9, 13, 34, 36 und 37 gefiltert. Nur diese Nachrichtentypen enthalten die eigentlichen Messwerten. Anschließend werden die IEC104-Nachrichten in ein Format für Messwerte gebracht, wie es im gesamten dashPORT-

<sup>8</sup> <https://git.swl.informatik.uni-oldenburg.de/projects/OJ/repos/oj104/browse>

System verwendet wird. Da IEC 104 nur eine binäre Information Object Address (IOA) übermittelt und nicht die intern verwendeten MeterId findet in diesem Schritt auch ein Mapping der IOA auf MeterId statt. Das Mapping ist einer beigelegten CSV abgelegt. Bevor die Messwerte weiter an RabbitMQ gesendet werden, findet noch eine Behandlung möglicher Fehler statt. Dazu gehört beispielsweise Fehler bei der Skalierung der Werte. Erst dann werden die Messwerte im JSON-Format an RabbitMQ gesendet.

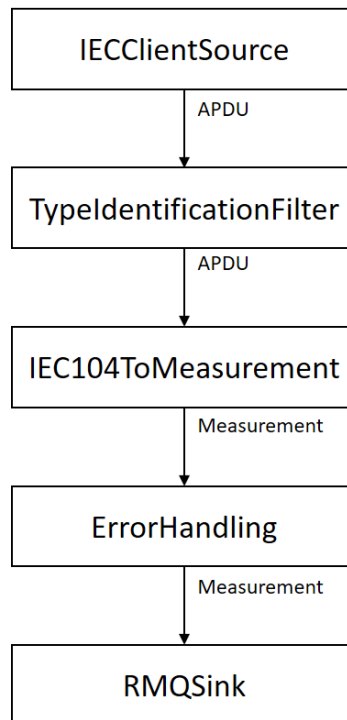


Abbildung 17: Anfrageplan IEC104-Adapter

## OPC-Adapter

Der OPC-Adapter verfolgt einen pull-basierten Ansatz. Er nimmt eine Verbindung zum OPC-Server von J.Müller auf um eine vorgegebene Liste von Messpunkten abzurufen. Ein Filtern und Mapping wie es beim IEC-104-Adapter gemacht wird, ist nicht nötig. Bevor die Messwerte weiter an RabbitMQ gesendet werden, wird eine Fehler-Behandlung durchgeführt.

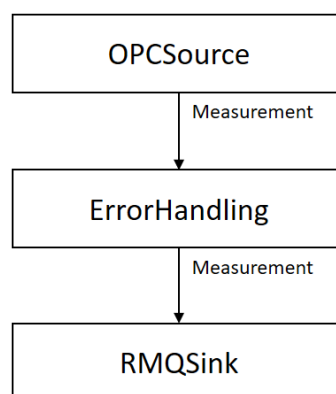


Abbildung 18: Anfrageplan OPC-Adapter

## Umsetzung der Prognose und Handlungsempfehlungen

Die Implementierung der Prognosemodelle erfolgte ausschließlich in Python. Für Python existieren viele etablierte Frameworks, die für die Datenvorverarbeitung, zum Modelltraining oder zur Visualisierung eingesetzt werden können, weshalb es sich für die Entwicklung prototypischer KI-Tools eignet.

### Datenvorverarbeitung

Für die Datenvorverarbeitung wird zwischen der Vorverarbeitung für das Training und der Datenvorverarbeitung im Echtbetrieb unterschieden. Der Grund dafür ist, dass für das Training eine große Menge an Daten vorbereitet werden muss, die sowohl Eingabe, als auch Ausgabedaten enthalten, anhand derer die Modelle die Lastverläufe lernen können. Im Echtbetrieb müssen aktuelle Daten für die Eingabe in die Modelle ausgelesen und vorverarbeitet werden. Nachdem die Prognose durchgeführt wurde ist außerdem eine Nachbearbeitung der Ausgabedaten für die Speicherung und die Anzeige im Dashboard notwendig. Grundsätzlich wurde für die Vorverarbeitung das Framework *pandas*<sup>9</sup> eingesetzt, welches es erlaubt tabellarische Daten zu verarbeiten.

Für die Trainingsdaten wurde zunächst eine Abfrage der Daten aus der Datenbank durchgeführt. Dadurch können die Rohdaten, die erfasst wurden ausgelesen werden. In der Datenbank wurden hier bereits soweit möglich fehlende Datenpunkte durch den letzten vorherigen bekannten Wert aufgefüllt. Da aber teilweise auch danach noch Datenpunkte fehlen, beispielsweise, wenn zu Beginn des Zeitraums Daten fehlen, wurden diese anschließend in einem Python Skript mit Hilfe von *pandas* aufgefüllt. Danach werden die Zeitreihen in Trainingsdaten umgewandelt, indem Eingabe- und Ausgabepaare gebildet werden. Dazu werden vier aufeinanderfolgende Datenpunkte als Eingabedaten, und die nächsten vier aufeinanderfolgende Datenpunkte als Ausgabedaten erfasst. Dieser Schritt wird für jeden Zeitpunkt im betrachteten Zeitraum wiederholt, sodass ein Datensatz entsteht, in dem jeder Datenpunkt die Werte der letzten Stunde als Eingabe und die Werte der nächsten Stunde als Ausgabe enthält.

Für das Training müssen die Daten in 80% Trainings- und 20% Testdaten unterteilt werden. Die Testdaten sind dabei dazu da, die Prognosegüte auf Daten zu evaluieren, die das Modell nicht zum Training verwendet hat. Durch diese Evaluation kann eine bessere Aussage über die Leistung im Echtbetrieb getätigt werden. Außerdem werden die Daten für das Training zwischen 0 und 1 skaliert, da die Aktivierungsfunktionen, welche in künstlichen neuronalen Netzen angewendet werden, nicht mit beliebigen Wertebereichen funktionieren. Diese Skalierer werden gespeichert, da die selben Skalierungen auch im Echtbetrieb genutzt werden müssen, um die Eingabedaten in die korrekte Skalierung zu bringen und die Ausgabedaten wieder in die tatsächlichen Wertebereiche zurückzuführen.

Das Training der Modelle erfolgt mithilfe des KI-Frameworks *keras*.<sup>10</sup> Keras enthält viele vorimplementierte Modelltypen, die sich in so genannten Layern beliebig miteinander kombinieren lassen. Außerdem sind viele Trainingsalgorithmen bereits enthalten. Es ermöglicht das Training auf Grafikbeschleunigern, was, die richtige Hardware vorausgesetzt, das Training deutlich beschleunigt. Für die Prognosemodelle wurden Modelle mit einem LSTM Layer mit 8 Knoten, sowie ein Dense Layer mit 4 Knoten für die Ausgabe verwendet. Diese wurden mit dem *adam* Optimierer mit Mean Squared Error (MSE) loss, einer Lernrate von 0.001 und einer batch size von 16 über 200 Epochen trainiert. Im Anschluss an das Training wurden die Daten in ihren ursprünglichen Wertebereich zurückskaliert um die Prognosegüte auszuwerten. Zum Schluss werden die trainierten Modelle abgespeichert um sie im Echtssystem nutzbar zu machen.

---

<sup>9</sup> <https://pandas.pydata.org/>

<sup>10</sup> <https://keras.io/>

## Deployment

Für das Deployment, also den Betrieb im Echtsystem, werden die Prognosemodelle in einem *Docker*<sup>11</sup> Container ausgeführt. Dadurch sind Anpassungen leicht möglich und die Prognose läuft abgekapselt von anderen Prozessen auf dem gleichen System. Alle Prognosen werden dabei in einem einzigen Container ausgeführt.

Zunächst wird eine Verbindung zu RabbitMQ aufgebaut, auf dem die Datenströme der live-Daten verwaltet werden. Hier werden für jede Einspeisung die neuen eintreffenden Messwerte empfangen und zwischengespeichert. Immer zur vollen Stunde, 15 Minuten danach, 30 Minuten danach und 45 Minuten danach werden die zwischengespeicherten Werte vorverarbeitet, damit sie in die Prognosemodelle eingegeben werden können. Dazu werden sie zunächst aggregiert und in das richtige Eingabeformat gebracht. Anschließend werden die Eingabedaten mit dem im Training gespeicherten Skalierer skaliert und die Prognose durchgeführt. Die Ausgabe wird analog zur Eingabe zurückskaliert und dann wieder an RabbitMQ übergeben, welches die Prognose dann an die Datenbank und das Dashboard weiterleitet. Wird eine Überschreitung der Lastspitze in der Prognose festgestellt, wird die Handlungsempfehlungslogik ausgeführt, welche im selben Container implementiert wurde.

Für die geplante Prognose der einzelnen Verbraucherleistungen vor der Durchführung der Handlungsempfehlungslogik wurden Daten zu Schiffpositionen benötigt. Diese können über eine von NPorts installierte AIS Antenne bereitgestellt werden. Alle Schiffe sind mit einem Transponder ausgestattet, mit dem sie regelmäßig eine Nachricht mit ihrer Position verschicken. Im selben Container, in dem die Prognose und Handlungsempfehlung ausgeführt wird, werden daher auch die Daten der Antenne empfangen und zwischengespeichert. Da bestimmt werden soll, in welchem Hafenbereich wie viele Schiffe liegen, werden die eingehenden Nachrichten zunächst gesammelt. Alle 5 Minuten werden die gesammelten Nachrichten konsolidiert und mittels Geofencing bestimmt, in welchem der sieben Hafenbereiche sich in diesen 5 Minuten wie viele Schiffe befunden haben. Diese Daten werden gespeichert und können bei Bedarf vom Prognose- oder Handlungsempfehlungsmodul ausgelesen werden.

### 2.2.6.4 Umsetzung der grafischen Benutzeroberfläche

Die Umsetzung der grafischen Benutzeroberfläche erfolgte in Form einer Web-Applikation (vgl. Abbildung 19), welche mithilfe der Technologien Express, React und MongoDB verwirklicht wurde. Frontend und Backend der Web-Applikation wurden vollständig in Javascript implementiert.

---

<sup>11</sup> <https://www.docker.com/>

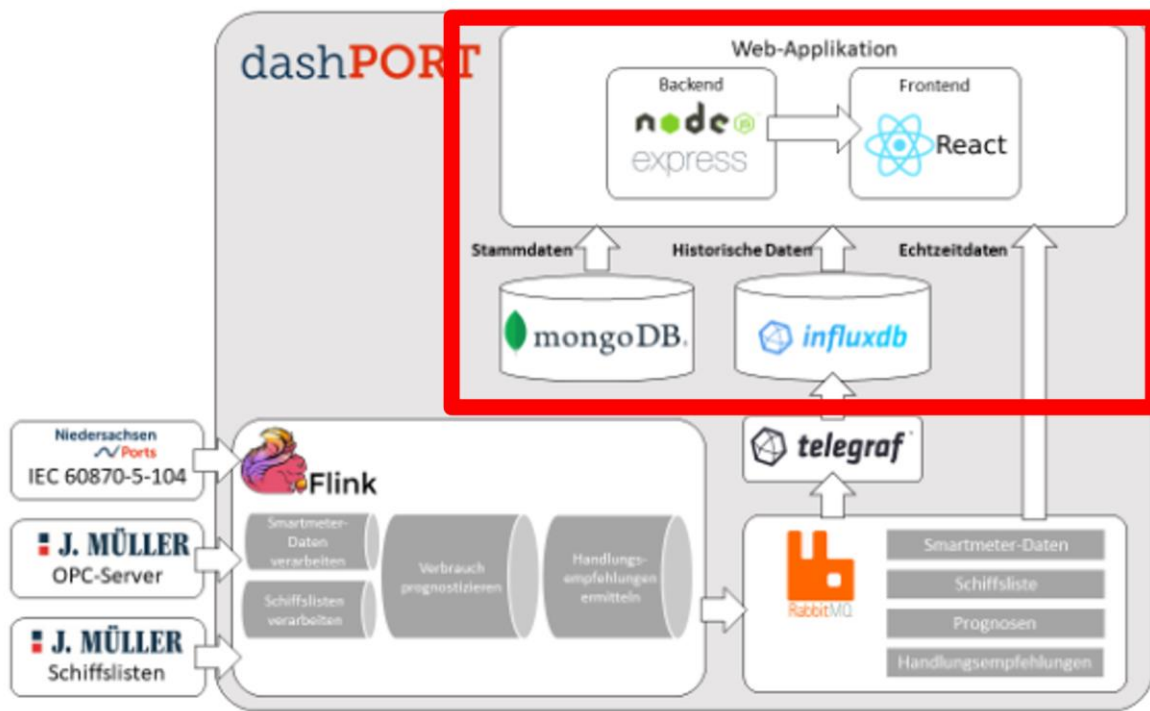


Abbildung 19: Softwarearchitektur von DashPORT mit Fokus auf die Web-Applikation

Nachfolgend werden die für die Umsetzung der grafischen Benutzeroberfläche verwendeten Technologien vorgestellt und ihr Verwendungszweck kurz erläutert.

## Backend mit Express.js

Express.js<sup>12</sup> ist ein modernes und schnelles auf Javascript basierendes Node.js-Backend-Framework. Es bietet viele Funktionen für die Entwicklung skalierbarer Backend-Anwendungen.

Im Projekt wurde express als Backend im Frontend-Bereich verwendet und bildet somit die Schnittstelle zwischen den Datenbanksystemen InfluxDB und MongoDB und dem Frontend.

Das Backend ist dafür zuständig Anfragen (en.: Requests) von der Benutzeroberfläche über bereitgestellt Endpunkte entgegenzunehmen, die angefragten Daten von den InfluxDB- und MongoDB-Diensten einzuholen und aufbereitet an das Frontend zurückschicken. Eine vollständige Liste der vom Frontend verwendeten Endpunkte findet sich nachstehend in Tabelle 6: Auflistung der vom Frontend verwendeten Endpunkte .

Endpunkte „SMARTMETER“	
GET SMARTMETER	Gibt die Zeitreihendaten eines Smartmeters für einen gewünschten Zeitraum in einer gewünschten Auflösung (Minuten, Stunden, Tage) zurück.
GET META	Gibt die in der MongoDB gespeicherten Metadaten wie Bezeichnung, Beschreibung, Standort für ein gewünschtes Smartmeter zurück.
GET POWERSUPPLIES	Gibt eine Liste aller Einspeisungen mit den zugehörigen Smartmetern zurück.
GET PREDICTION	Gibt die Prognosedaten eines Smartmeters für die nächste Stunde in 15 Minuten-Intervallen zurück

<sup>12</sup> <https://expressjs.com/>

GET JUNCTIONS	Gibt eine Liste aller Verteiler mit den zugehörigen Smartmetern zurück.
GET CLUSTERS	Gibt eine Liste aller Verbrauchscluster mit den zugehörigen Smartmetern zurück.
GET PEAK	Gibt den Wert der höchsten Lastspitze einer Einspeisung zurück.
GET LAST	Gibt den letzten Messwert eines Smartmeters zurück.
<b>Endpunkte „RECOMMENDATIONS“</b>	
GET RECOMMENDATIONS	Gibt die Handlungsempfehlungen zu einer Einspeisung zurück.
POST ACTIVE	Setzt angenommene Vorschläge einer Handlungsempfehlung und die Handlungsempfehlung selbst auf „aktiv“.
<b>Endpunkte „USER“</b>	
GET LOGGED_IN	Gibt zurück, ob der anfragende Nutzer bereits eingeloggt ist.
POST LOGIN	Nimmt die Login-Daten entgegen, prüft die Daten und ermöglicht dem Nutzer sich am System anzumelden.
POST LOGOUT	Loggt den Nutzer aus dem System aus.
<b>Endpunkte „STATUS“</b>	
GET MONGO_STATUS	Gibt zurück, ob der MongoDB-Service erreichbar ist.
GET INFLUX_STATUS	Gibt zurück, ob der InfluxDB-Service erreichbar ist.

Tabelle 6: Auflistung der vom Frontend verwendeten Endpunkte

## Frontend- und UI-Entwicklung mit React.js

Bei React.js<sup>13</sup> handelt es sich um eine moderne Javascript-Bibliothek, welche für die komponentenbasierte Entwicklung von Benutzeroberflächen verwendet wird. In DashPORT wurde die gesamte Benutzeroberfläche mit React.js in Javascript entwickelt. Zu diesem Zwecke wurden alle identifizierten Bereiche der Benutzeroberfläche in separate und wiederverwendbare Komponenten zerlegt und weitestgehend isoliert voneinander implementiert.

## Nutzer- und Metadatenhaltung mit MongoDB

Bei MongoDB<sup>14</sup> handelt es sich um ein schnelles, nicht-relationales NoSQL-Datenbanksystem, das für die Speicherung großer Datenmengen verwendet werden kann.

Im Projekt wurde MongoDB verwendet, um Login-Informationen und Metadaten zu speichern. Zu den Metadaten zählen alle Informationen, die für jedes Smartmeter entweder zum Abfragen weiterer Informationen aus der InfluxDB oder zur Darstellung im Frontend benötigt wurden. Dazu zählen Informationen wie der Standort des Smartmeters, die Bezeichnung, die Beschreibung, Abkürzungen. Eine vollständige Liste der mittels MongoDB verwendeten Tabellen mit beispielhaften Datensätzen findet sich nachstehend in Tabelle 7.

In der Tabelle „*meters*“ befinden sich die bereits beschriebenen Metadaten der jeweiligen Smartmeter. In „*recommendations*“ befinden sich die Datensätze der Handlungsempfehlungen. In „*users*“ die Datensätze der Nutzer und in „*ships*“ die importierten Daten aus der Schiffsliste.

Tabellenbezeichnung	Beispielhafter Datensatz
---------------------	--------------------------

<sup>13</sup> <https://reactjs.org/>

<sup>14</sup> <https://www.mongodb.com/>

<i>meters</i>	<pre> meterId: "X-S-01-01" __v: 0 abbreviation: "AX_EZ1_J01" area: "Energiezentrale EZXNordpier" cluster: "Cluster - 0" consumer: "20kV-Anlage EZ1, X01 Einspeisezelle" description: "Ringleitung Hafenversorgung" gridElement: "Einspeisung Mittelspannung" hierarchy: "0" junction: "0" measureDevice: "UMG 507" organization: "np" powerSupply: "0" structure: "Althafen" </pre>
<i>recommendations</i>	<pre> powersupply: "sm_4711" time: 2022-06-30T17:02:42.000+00:00 ✓ recommendations: Object   ✓ 1: Object     AH_HV13_V2_P6: 32.493690490722656     total: 32.493690490722656   &gt; 2: Object   &gt; 3: Object   ✓ 4: Object     AH_HV12_V2_P20: 2.9592819213867188     AH_HV12_V4_P17: 2.3184914588928223     AH_HV14_V3_P6: 6.48093843460083     total: 11.758711814880371   &gt; 5: Object   &gt; 6: Object </pre>
<i>users</i>	<pre> _id: ObjectId('6278d99334f4edblaa4cce61') username: "nports" passwordHash: "\$2a\$10\$zFXithTXvXQrJoNBxwZv0uPkMP/W.hVqmwda8gj company: "np" __v: 0 </pre>
<i>ships</i>	<pre> name: "Rostrum Europe" kind: "Seeschiff" ✓ eta: Object   start: 2021-10-06T21:30:00.000+00:00   end: 2021-10-06T21:30:00.000+00:00 ✓ planning: Object   poller_start: "0.0"   poller_end: "0.0"   landseite: ""   anlegedatum: ""   anlegezeit: ""   planungszusatz: ""   raw_text: "Von Poller: 0.0 Bis Poller: 0.0 Landseite: An ✓ goods: Array   ✓ 0: Object     amount: "35510"     kind: "Schnittholz (in cbm)" </pre>

Tabelle 7: Verwendete Tabellen der MongoDB

## Kommunikation zwischen den Komponenten

Die Kommunikation zwischen dem Backend und seinen vor- und nachgelagerten Komponenten wurde an die RESTful-Architektur angelehnt. Das Frontend fragt beim Backend bestimmte Ressourcen an, diese werden dann wiederum vom Backend bei der InfluxDB oder MongoDB angefragt. Die Anfrageergebnisse werden aufbereitet und zurück ans Backend bzw. das Frontend geschickt. Abbildung 20 stellt beispielhaft die Kommunikation zwischen den Komponenten dar, wenn im Frontend eine Einspeisung angeklickt wird. Zunächst schickt das Frontend vier Anfragen an das Backend: Für die Smartmeter-Messdaten der letzten Stunde, für die Prognosedaten der kommenden Stunde, für die Metadaten und für die Handlungsempfehlungen. Das Backend holt sich die entsprechenden Daten von der InfluxDB und MongoDB ein und schickt sie zurück ans Frontend. Sind im Frontend alle Daten eingegangen, wird die Oberfläche gerendert und der Nutzer sieht das Ergebnis.

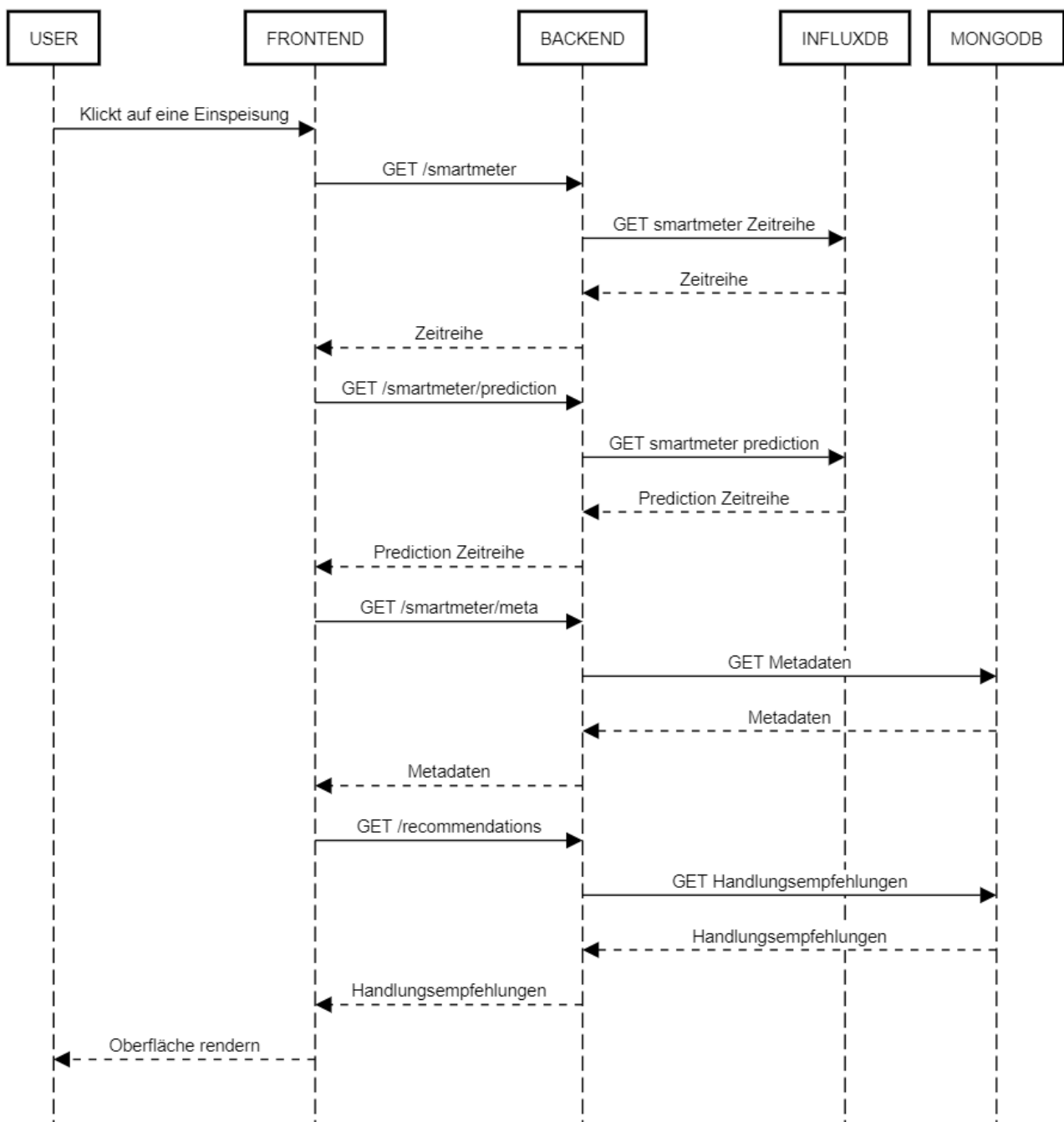


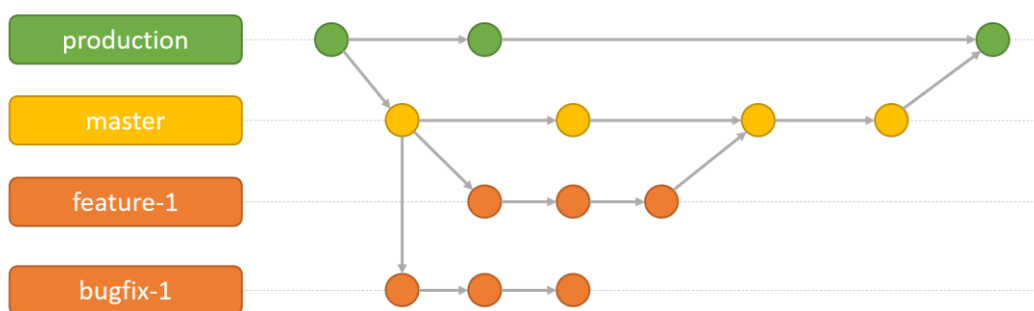
Abbildung 20: Sequenzdiagramm "Nutzer klickt auf eine Einspeisung"



## 2.2.7 (TAP 6.3) Tests dashPORT

Bei der Entwicklung des dashPORT-Systems wurde ein Continuous Integration-Ansatz (CI) verfolgt. Dadurch war es möglich zeitnah Fehler am System zu erkennen und entgegenzuwirken. Hierfür wurden, die in Abbildung 21: Deployment-Umgebungen

Neue Features und Bugfixes werden in einem Feature-Branch entwickelt und können bei Bedarf über eine Gitlab-Pipeline in der lokalen OFFIS Test Environment deployt werden. Sobald die Entwicklung an einem Feature-Branch beendet ist, wird er nach einem Review in den master-Branch gemergt. Der master-Branch wird nach einem Merge automatisch von einer Gitlab-Pipeline in der J. Müller Test Environment deployt. Über diese Umgebung können Projektpartner neue Features testen und es können vom OFFIS Integrationstests durchgeführt werden. Durch das unmittelbare aktualisieren des Systems nach Änderungen, werden Fehler erkannt, bevor Änderungen am Produktivsystem vorgenommen werden.



dargestellten,

Umgebungen für das Deployment des Systems bereitgestellt. Des Weiteren wurde der Git-Workflow mit Feature-Branche verwendet.

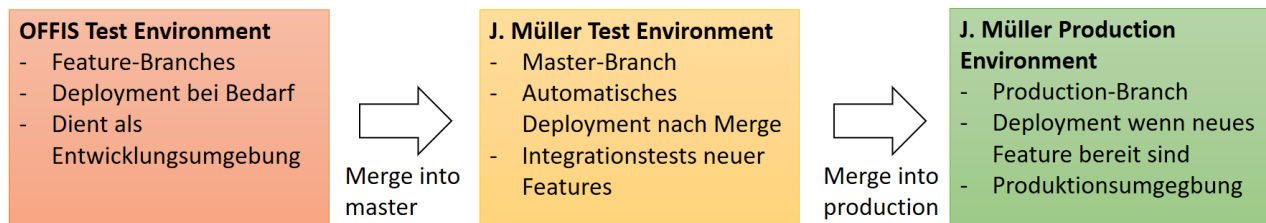


Abbildung 21: Deployment-Umgebungen

Neue Features und Bugfixes werden in einem Feature-Branch entwickelt und können bei Bedarf über eine Gitlab-Pipeline in der lokalen OFFIS Test Environment deployt werden. Sobald die Entwicklung an einem Feature-Branch beendet ist, wird er nach einem Review in den master-Branch gemergt. Der master-Branch wird nach einem Merge automatisch von einer Gitlab-Pipeline in der J. Müller Test Environment deployt. Über diese Umgebung können Projektpartner neue Features testen und es können vom OFFIS Integrationstests durchgeführt werden. Durch das unmittelbare aktualisieren des Systems nach Änderungen, werden Fehler erkannt, bevor Änderungen am Produktivsystem vorgenommen werden.

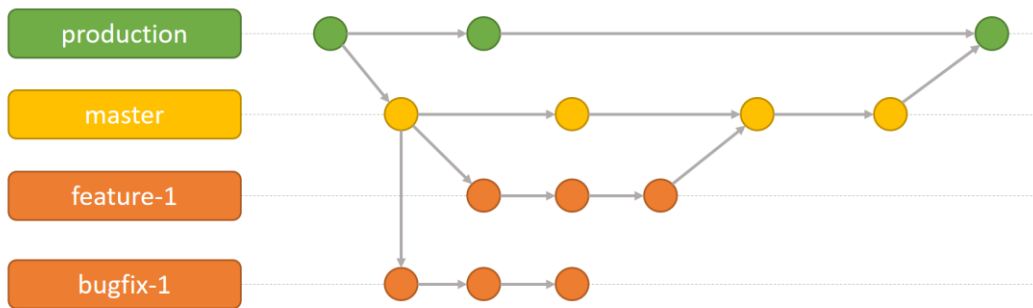


Abbildung 22: Git-Workflow

Sobald im Master-Branch wesentliche Fortschritte gemacht wurden, wird dieser in den production-Branch gemergt und eine Gitlab-Pipeline zum Deployment des Systems auf der Produktivinfrastruktur gestartet.

Neben den Pipelines für das Deployment gibt es Build-Pipelines für Softwareartefakte. Die Build-Pipelines werden nach jeder Änderungen am Code in allen Branches durchgeführt. Nach jeder Build-Pipelines wird ein Smoke-Test durchgeführt um zu überprüfen, ob das System nach den Änderungen noch funktionsbereit ist.

## 2.2.8 AP 7: Feldtest im Hafen

Der Feldtest wurde von Fraunhofer CML geleitet. OFFIS hat in Teilarbeitspaket 7.1 die Integration des Gesamtsystems auf der IT-Infrastruktur des Hafens durchgeführt.

OFFIS hat die Installation der Softwarekomponenten auf der IT-Infrastruktur des Hafens durchgeführt und war für die Durchführung von Funktionstests der Software und deren Schnittstellen in Abstimmung und Zusammenarbeit mit NPorts und J. Müller verantwortlich.

Außerdem hat OFFIS in Teilarbeitspaket 7.2, gemeinsam mit Fraunhofer CML, die Mitarbeiter des Hafens auf der entwickelten Software geschult und in Teilarbeitspaket 7.3 die Funktionalität der Software im Echtbetrieb getestet. Gefundene Fehler wurden umgehend behoben und im Echtbetrieb erprobt.

## 2.2.9 AP 8: Evaluierung und Verwertung

Die Evaluierung und Verwertung wurde von Fraunhofer CML geleitet. OFFIS hat Feedback erhoben und umgesetzt, soweit die Maßnahmen von Fraunhofer CML auf technischer Seite unterstützt, indem beispielsweise gewünschte Änderungen im Gesamtsystem umgesetzt wurden.

Im Anschluss an die Testphase wurden qualitative Interviews mit den Nutzern von dashPORT durchgeführt. Die Handlungsempfehlungen wurden als hilfreich bewertet und besonders die Benutzerfreundlichkeit der Benutzeroberfläche wurde hervorgehoben.

Das detaillierte Ergebnis der Evaluierung findet sich im Gesamtbericht.

## 2.3 Verwendung der Zuwendung

Die Zuwendung wurde fast ausschließlich für Personalkosten verwendet. Die Akquise von Wissenschaftlichen Hilfskräften war nicht erfolgreich, was zu Einsparungen, aber auch einem

höheren Aufwand bei den Wissenschaftlichen Mitarbeitern geführt hat. Außerdem gab es Corona-bedingte Einsparungen bei den Dienstreisen, welche genutzt wurden, um kostenneutral die höheren Aufwände anteilig in anderen Positionen decken zu können. Zudem gab es eine kostenneutrale Verlängerung von insgesamt 6 Monaten. Insgesamt ist der Kostenplan eingehalten worden.

## 2.4 Notwendigkeit der geleisteten Arbeit

Die Installation der notwendigen Infrastruktur und Software erforderte hohe finanzielle Aufwendungen. Obwohl das dashPORT-System theoretisch hohe Einsparungen ermöglicht, war es vor Projektbeginn schwierig, diese genau zu quantifizieren. Diese Einsparungen hängen von einer Reihe von Maßnahmen ab, die im Rahmen des Projekts umgesetzt werden mussten. Eine wesentliche Energieeinsparung ergibt sich aus dem Datenaustausch zwischen den verschiedenen Akteuren im Hafen, der nur durch einen sicheren, unternehmensübergreifenden Austausch von Daten ermöglicht wird. Vor Beginn des Projekts gab es jedoch keine technische Grundlage oder Erfahrungswerte für einen solchen Austausch zwischen den Hafenakteuren. Aus diesem Grund war es notwendig, das Projekt in ein Forschungsvorhaben zu integrieren.

Ein weiteres Risiko bestand darin, dass Flexibilisierungsmaßnahmen und Lastspitzenverschiebungen möglicherweise nicht mit den bestehenden Hafenprozessen vereinbar sind. Eine Untersuchung hierzu hatte im Braker Hafen noch nicht stattgefunden und war daher ein notwendiger Bestandteil des Projekts. Die Förderung ermöglichte eine strukturierte Projektdurchführung über mehrere Jahre hinweg, deren langfristige Auswirkungen auf die Kostenstruktur im Hafen erst in der Zukunft spürbar sein werden. Die Zusammenarbeit mit motivierten Partnern aus Wissenschaft und Industrie war eine Herausforderung, die ohne Fördermittel auf allen Seiten kaum bewältigt werden konnte, insbesondere aufgrund des langen Projektzeitraums. Die Forschungsinstitute Fraunhofer CML und OFFIS sind als Einrichtungen der öffentlichen Hand auf die Projektförderung angewiesen. Ohne eine solche Förderung können geplante Projekte nicht weiterverfolgt werden. Die Zusammenarbeit mit industriellen Projektpartnern erhöhte die Zukunftsausrichtung der Universitäten und die europäische sowie weltweite Wettbewerbsfähigkeit sowohl des wissenschaftlichen als auch des industriellen Standorts Deutschlands. Das OFFIS hatte vor der Antragstellung andere Finanzierungsmöglichkeiten geprüft und konnte keine alternativen Fördermöglichkeiten identifizieren.

## 2.5 Verwertbarkeit

Die Software wurde so konzipiert, dass sie grundsätzlich parametrisierbar und anpassbar ist. Das ermöglicht es, sie bei vorhandener Messinfrastruktur und daraus resultierender Datensammlung, das System auch an anderen Standorten zu betreiben.

OFFIS ist ein Forschungsinstitut welches keine eigenen wirtschaftlichen Interessen aus der Forschung verfolgt. Daher sieht sich OFFIS vor allem als Schnittstelle zwischen Forschung und Anwendung, um wirtschaftsbezogene Firmen die Möglichkeit zu geben Ergebnisse aus der Forschung in die wirtschaftliche Anwendung zu übertragen. Das erworbene Wissen und die Tätigkeiten innerhalb des Projektes lassen sich somit mit der Verwertungsperspektive der industriellen Partner NPorts und J.Müller in die Wirtschaft hinaustragen.

Auf wissenschaftlich technischer Ebene haben sich kurz nach Projektabschluss bereits einige Verwertungsmöglichkeiten ergeben. Das Interesse an den Ergebnissen des dashPORT Projekts

innerhalb von OFFIS ist groß und die Verwendung der Software oder Teilen der Software wird bereits in mehreren anderen Forschungsprojekten evaluiert. Nach Projektende wird die Software als Open Source Software veröffentlicht, was es jedem ermöglicht, diese sowohl in anderen Forschungsprojekten, als auch in der Wirtschaft einzusetzen. Dadurch ist gewährleistet, dass die Ergebnisse vielfältig weiterverwendet werden können.

Im Projekt dashPORT wurde gezeigt, dass es möglich ist ein Monitoring-System von Stromverbrauchsdaten, welches ebenfalls die Möglichkeit bietet, die Verbräuche zu prognostizieren, sowie Handlungsempfehlungen zur Vermeidung von Lastspitzen zu geben, zu entwickeln. In weiteren Forschungsprojekten könnte die Übertragbarkeit der Methodik und der Technologie auf andere Domänen und andere Energieträger untersucht werden. Das Datenstrommanagementsystem, welches der Software zugrunde liegt, könnte als Grundlage für Software in anderen Forschungsprojekten dienen. Besonders in Produktionsbetrieben, die planbare Prozessabläufe haben, könnten die in dashPORT entwickelten Prognosesysteme einen erhöhten Mehrwert bieten.

## 2.6 Fortschritt anderer Stellen

Innerhalb der Projektlaufzeit wurde ein kontinuierlicher Abgleich mit dem Stand der Wissenschaft durchgeführt. Bei verschiedensten Möglichkeiten, wie z.B. Konferenzen, Forschungsaustausch, Diskussionen mit Industrievertretern, Fachvorträgen, Patentrecherchen usw., wurden keine relevanten Fortschritte im Themengebiet des Forschungsvorhabens festgestellt.

## 2.7 Veröffentlichungen

Zur Projektlaufzeit sind keine wissenschaftlichen Veröffentlichungen seitens OFFIS entstanden.

Über das Projekt wurde regelmäßig auf der OFFIS Homepage, sowie in der Hauszeitschrift Datawork (Ausgabe 74/07.2022) informiert. Innerhalb des OFFIS wurde das Projekt und seine Ergebnisse auf mehreren Veranstaltungen präsentiert und somit für Forscher anderer Domänen zugänglich gemacht.

Zusätzlich wurde dashPORT auf externen Veranstaltungen durch OFFIS repräsentiert:

- Vorstellung der vorläufigen Projektergebnisse bei der Lübecker Hafengesellschaft, 01/2021
- MCN-Cup 2021: Einreichung und Ehrung mit erstem Preis in der Kategorie Nachhaltigkeit in Häfen
- SMM, Hamburg, 2022, Stand des Maritimen Clusters Norddeutschland
- IHATEC Vernetzungskonferenz 2022, Stand und Kurzvorträge
- Vorstellung der Projektergebnisse in gemeinsamem Workshop mit der Hamburg Port Authority und Airbus Hamburg, 2022

Geplante Veröffentlichungen, Vorträge und Workshops:

- Vortrag bei Port of Sevilla, Datum ausstehend.

## 3 Abbildungsverzeichnis

Abbildung 1: Gantt Chart dashPORT .....	4
Abbildung 2: Systemkontextsicht auf dashPORT .....	4
Abbildung 3: Gegenüberstellung der Einspeisungen AH_EZ1_J01 (orange) und AH_HV11_V3_P1 (blau) .....	4
Abbildung 4: Lambda-Architektur .....	5
Abbildung 5: Kappa-Architektur.....	6
Abbildung 6: Softwarearchitektur der digitalen Leitwarte .....	7
Abbildung 7: DashPORT Entwurf "Startseite" .....	8
Abbildung 8: DashPORT Entwurf "Startseite" – Überblick über Bereiche .....	8
Abbildung 9: Anpassung der Benutzeroberfläche an angenommene Handlungsempfehlungen .....	8
Abbildung 10: dashPORT Benutzeroberfläche "Verbrauchsabschätzung" .....	9
Abbildung 11: dashPORT Benutzeroberfläche „Analyse“ .....	9
Abbildung 12: dashPORT Benutzeroberfläche "Prognose" .....	10
Abbildung 13: Mobile Ansicht der Benutzeroberfläche .....	10
Abbildung 14: dashPORT Benutzeroberfläche "Detailansicht" .....	10
Abbildung 15: Datenstruktur InfluxDB .....	11
Abbildung 16: Flink-Dataflow .....	12
Abbildung 17: Anfrageplan IEC104-Adapter .....	12
Abbildung 18: Anfrageplan OPC-Adapter .....	13
Abbildung 19: Softwarearchitektur von DashPORT mit Fokus auf die Web-Applikation .....	13
Abbildung 20: Sequenzdiagramm "Nutzer klickt auf eine Einspeisung" .....	13
Abbildung 21: Deployment-Umgebungen.....	15
Abbildung 22: Git-Workflow .....	18

## 4 Tabellenverzeichnis

Tabelle 1: Abgrenzung von verwandten Arbeiten .....	4
Tabelle 2: Beispielauszug Leistungswerte AH_ES1_J01 .....	4
Tabelle 3: Beispielauszug Zählerstand AH_ES1_J01 .....	4
Tabelle 4: Beispielauszug Leistungswerte P_MS_001 .....	5
Tabelle 5: Beispielauszug Zählerstände P_MS_001 .....	6
Tabelle 6: Auflistung der vom Frontend verwendeten Endpunkte .....	7
Tabelle 7: Verwendete Tabellen der MongoDB .....	8