



einfach. digital. agil.

PRÄSENTIERT

Proof of Concept - Migration auf ein Web Front-End mit IBM i Back-End

NOVEDAS

Software & Systems GmbH



einfach. digital. agil.

EIN BEITRAG VON

NOVEDAS

Software & Systems GmbH



Alexander Babel
Junior Developer



Matthias Wedig
Geschäftsführer

AGENDA

- Begrüßung
- Die Aufgabe
- Eine Lösung
- Node.js auf der IBM i
- Livedemo

Vorstellung: NOVEDAS Software & Systems GmbH

Softwareentwicklung	EDI	Projekt-Management
<ul style="list-style-type: none">▪ Passage- und Fracht-Buchungssysteme (IBM i, COBOL)▪ Check-In Systeme▪ Web-Client Buchungssysteme▪ Tracking und Tracing im Logistik-Bereich ▪ COBOL, Java, C++, Javascript▪ .Net▪ DB2▪ Security / Cryptography	<ul style="list-style-type: none">▪ Gefahrgutmeldungen▪ Interfaces zu Terminals▪ Zollverfahren▪ Web-Services Buchungssysteme ▪ XML▪ EDIFACT / UNICORN▪ ANSI X12	<ul style="list-style-type: none">▪ Projektleitung IT Entwicklung▪ Projekt-Controlling▪ Projektdurchführung▪ Requirements Engineering▪ Testmanagement

Die Aufgabe

Departure Date Time Ship Stat. Level
HRS THN 05.03.2016 15:00 NO B1 0

Name Book No. Cabin Registration

Groups: Status:
from to Checked Booked Passage Type
A A11 A11 A11

Pax Booked : 190 Checked : 191 SO : 0 Available : 1292

Name	Bookno	Chk	Book	Tp	Arrival
	35121556	Y	OK	FI	SFJ 08.03.21
	35844567	Y	OK	FI	SFJ 08.03.21
	35736465	Y	OK	F0	THN 07.03.21
	34891251	Y	OK	F0	THN 07.03.21
	35325061	Y	OK	F0	THN 07.03.21
	35865561	Y	OK	F0	THN 07.03.21
	35841064	Y	OK	F0	THN 07.03.21
	35609868	Y	OK	FI	SFJ 08.03.21
	35811666	Y	OK	FI	SFJ 08.03.21
	35525664	Y	OK	F0	THN 07.03.21
	35393265	Y	OK	FI	SFJ 08.03.21
	34686057	Y	OK	FI	SFJ 08.03.21

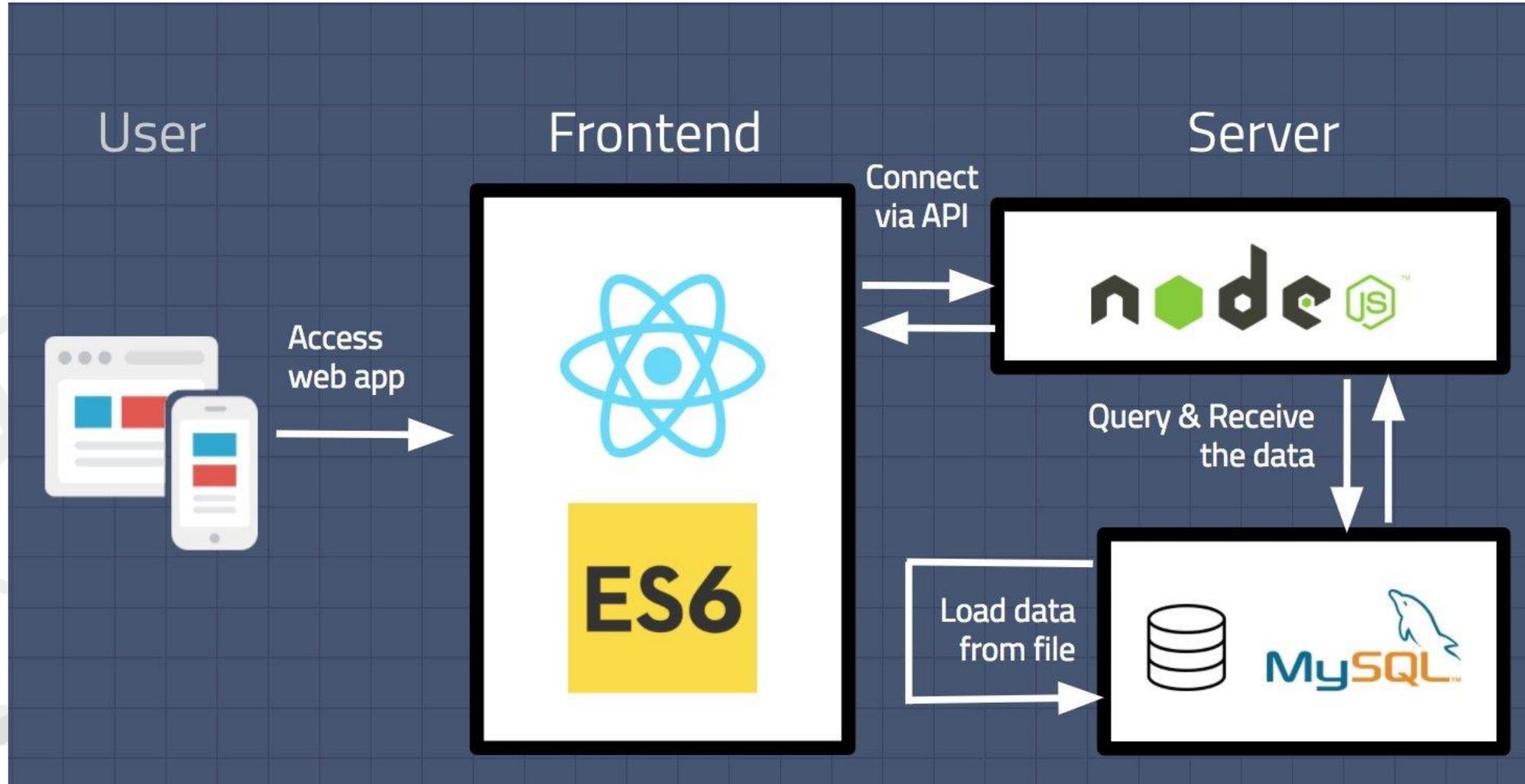
Migration der Check-In Komponente eines Passage Fährbuchungssystems

- Ablösung des VB6 Front-Ends durch ein modernes Web Front-End unter Beibehaltung des System i Back-Ends - Cobol Applikation und DB2

Ziele und Randbedingungen

- Entwicklung eines lauffähigen Proof-of-Concept mit den ausgewählten Web-Technologien
- Sichere Kommunikation und Authentifizierung musste erreicht werden
- System i Kernel durfte nicht verändert werden
- Wiederverwendbarkeit für andere, ähnliche Back-End Systeme auf der System i

Moderne Webarchitektur



Aufbau von Web Apps

Backend

- Datenbank: MySQL, DB/2, MongoDB
- Webserver: Node.js Server via Express.js
- API: REST, GraphQL

Frontend

- React.js: Virtual DOM, klares 1-Way Data Binding, alles Components
- Semantic-UI: CSS Framework mit voller React.js Integration

Node.js

Javascript Server

- Serverseitig: Für den serverseitigen Einsatz entwickelt
- Moderne Technik: Node.js läuft auf der JavaScript-Laufzeitumgebung „V8“ von Google
- JS Vorteile: JavaScript Eigenheiten schaffen Effizienz (e.g. non-blocking I/O)

Vielseitigkeit

- NPM: Node Package Manager als Paketmanager für Node.js mit Repo
- Module: Von Haus aus viele nützliche Module (e.g. „net“, „http“, „fs“)
- Erweiterbar: Mit Middleware wie „Express.js“ wird Node vielseitiger

Node.js

Javascript Server

- Serverseitig: Für den serverseitigen Einsatz entwickelt
- Moderne Technik: Node.js läuft auf der JavaScript-Laufzeitumgebung „V8“ von Google
- JS Vorteile: JavaScript Eigenheiten schaffen Effizienz (e.g. non-blocking I/O)

Vielseitigkeit

- NPM: Node Package Manager als Paketmanager für Node.js mit Repo
- Module: Von Haus aus viele nützliche Module (e.g. „net“, „http“, „fs“)
- Erweiterbar: Mit Middleware wie „Express.js“ wird Node vielseitiger

Node.js

Einfaches Beispiel eines HTTP-Servers

```
var http = require('http');  
  
http.createServer(function (req, res) {  
    res.writeHead(200, {  
        'Content-Type': 'text/html'  
    });  
    res.write('Hallo Welt');  
    res.end();  
}).listen(3000);
```

Express.js

Webserver noch einfacher

- Framework: “Express.js“ - Serverseitiges Webframework, das Web-Dev vereinfacht

```
// Wieder ein HTTP-Server
```

```
const express = require('express');
```

```
const app = express();
```

```
const server = app.listen(3000, function () {
```

```
  const host = server.address().address;
```

```
  const port = server.address().port;
```

```
  console.info(`Example app listening at http://${host}:${port}`);
```

```
});
```

```
const app = express() // Respond with "Hello World" when a GET request is made
```

```
app.get('/', function (request, response) {
```

```
  response.send('Hello World');
```

```
});
```

React.js

JavaScript Bibliothek für UI

- Bibliothek: Deklariert als Bibliothek, NICHT Framework
- Virtual DOM: Virtueller DOM „patched“ den eigentlichen DOM → schneller!
- JSX: Mix aus JS und XML. Einfach lesbar.
- Components: Alles einteilbar in kleinere wiederverwendbare Teile
- Data Binding: Props und States schaffen klaren Datenfluss

NOVEDAS Web Check-In Proof of Concept - Pakete

Server (Pakete)

- Express: Webservice & Routen
- JWT: JSON Web Token zur Realisierung eines Tokens
- XML-JS: XML Parsing
- XSD-Schema-Validator: Java basierte XML→XSD Validierung

Client (Pakete)

- Axios: Promise basierter HTTP Client
- React-DOM-Router: Routing Paket für React

NOVEDAS Web Check-In Proof of Concept - Features

Eingebaute Features

Server

- Login
- Usertoken
- Routen für verschiedene Funktionen
- XSD Validierung
- (TCP Verbindung zur IBM System i)

Client

- Token basiertes Ansprechen des Backends
- Asynchrone Abhandlung
- Zeitlimitiertes Sessionhandling (Safety)
- Dynamische Views

NOVEDAS Web Check-In Proof of Concept - Learnings

Was haben wir daraus gelernt?

Was können wir direkt übernehmen?

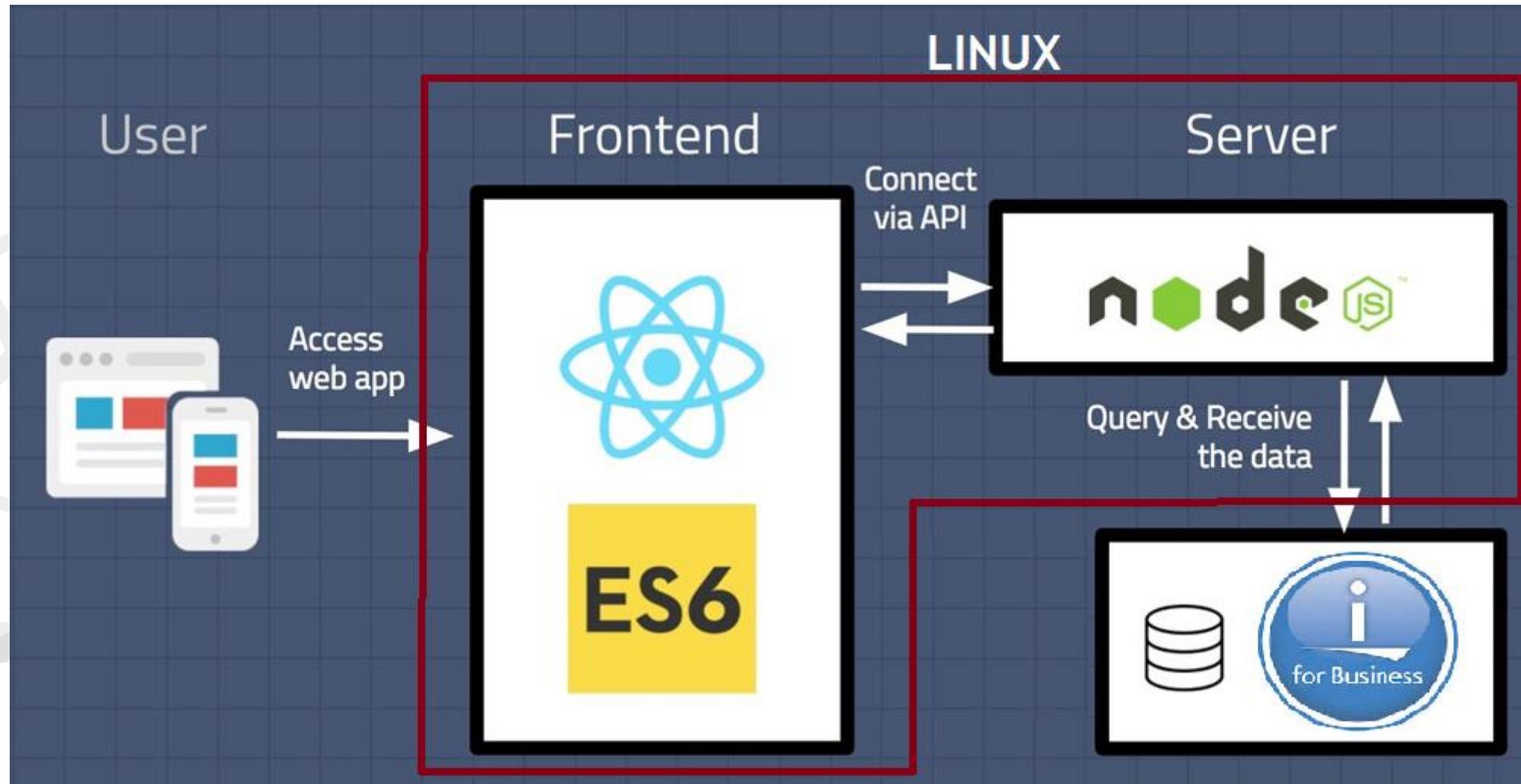
Client

- Navigation zwischen Views
- Kommunikation mit dem Backend
- Komponenten wie ‚Booking‘, ‚Bookinglist‘, ‚Login‘ haben durchaus multiple Einsatzorte

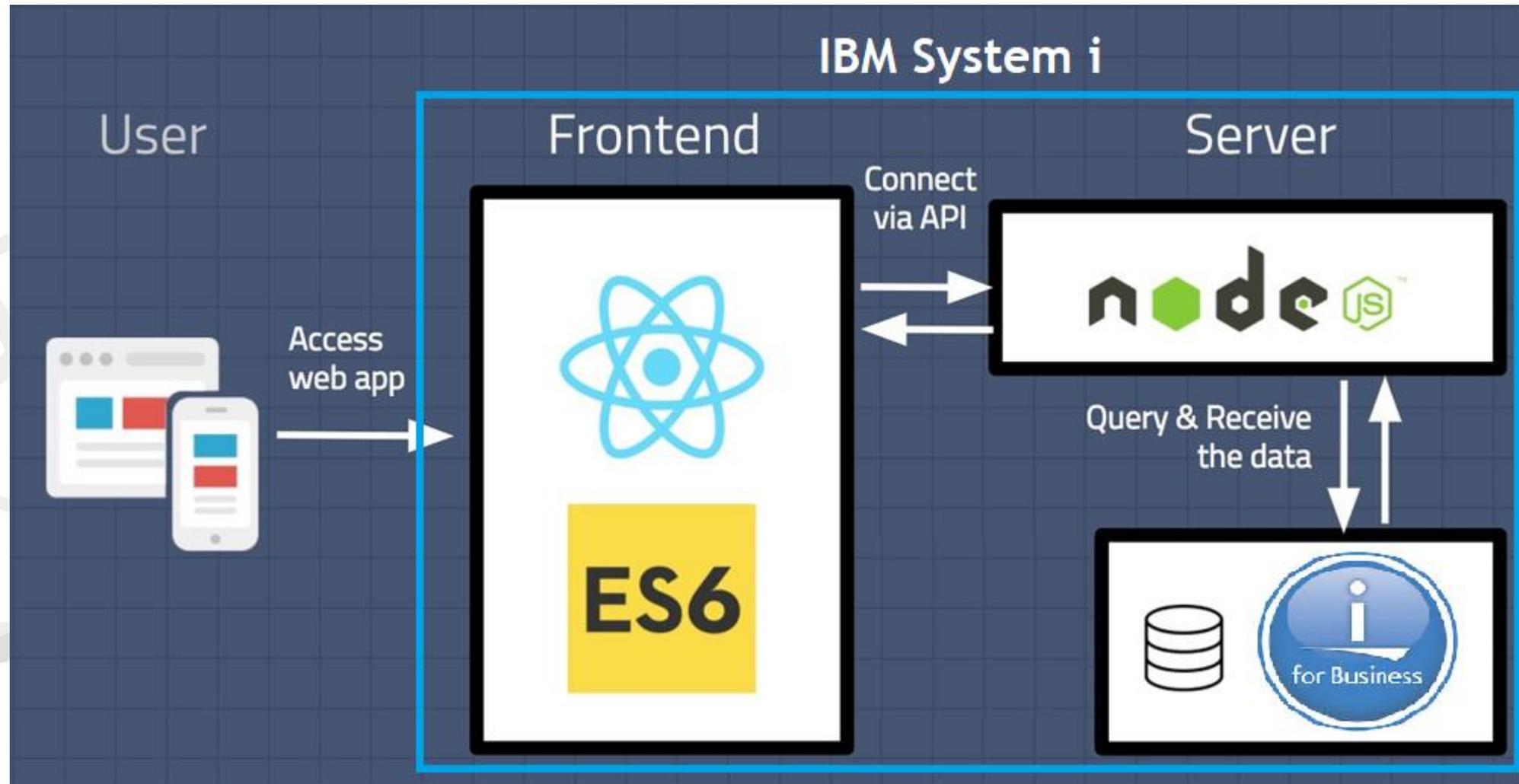
Server

- (System i Kommunikation)
- XML Verarbeitung
- XSD Validierung
- Tokenizing

Version 1 des POC - (Kein) Node.js auf der IBM i



Version 2 des POC - Node.js auf der IBM i

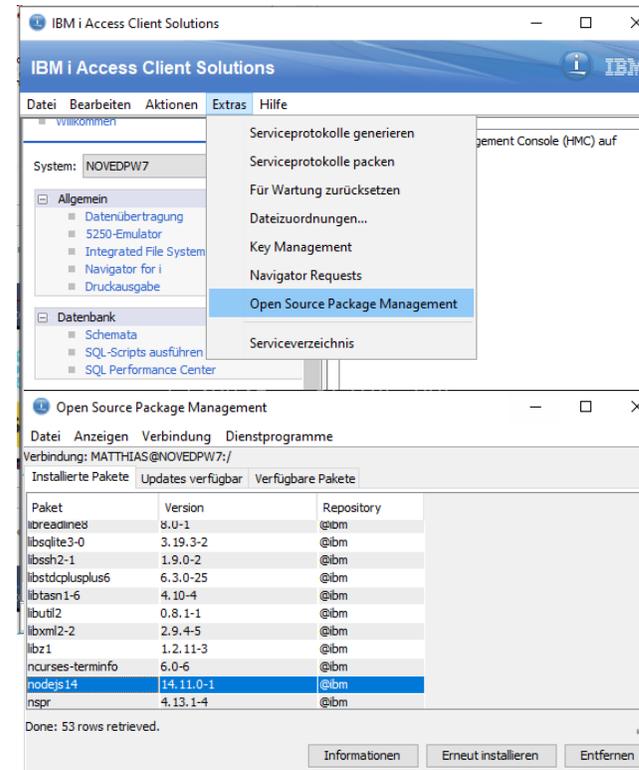


Node.js auf der IBM i - Installation

Voraussetzungen:

- Installiertes LPP 5733SC1 - IBM Portable Utilities for i
- SSH Zugang zur i:
 - STRTCPSVR *SSHD

Einfache Installation



The screenshot shows the IBM i Access Client Solutions interface. The top window is titled "IBM i Access Client Solutions" and has a menu bar with "Datei", "Bearbeiten", "Aktionen", "Extras", and "Hilfe". The "Extras" menu is open, showing options like "Serviceprotokolle generieren", "Serviceprotokolle packen", "Für Wartung zurücksetzen", "Dateizuordnungen...", "Key Management", "Navigator Requests", "Open Source Package Management" (highlighted), and "Serviceverzeichnis".

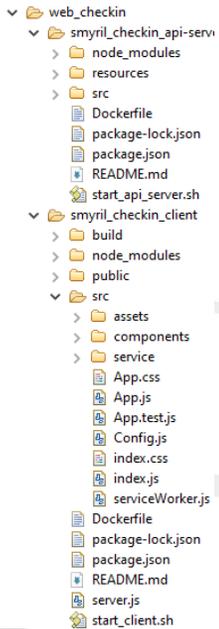
The bottom window is titled "Open Source Package Management" and shows a table of installed and available packages. The connection is identified as "MATTHIAS@NOVEDPW7:/".

Paket	Version	Repository
libreadline8	8.0-1	@ibm
libsquite3-0	3.19.3-2	@ibm
libssh2-1	1.9.0-2	@ibm
libstdcplusplus6	6.3.0-25	@ibm
libtasn1-6	4.10-4	@ibm
libutil2	0.8.1-1	@ibm
libxml2-2	2.9.4-5	@ibm
libz1	1.2.11-3	@ibm
ncurses-terminfo	6.0-6	@ibm
nodejs14	14.11.0-1	@ibm
nspr	4.13.1-4	@ibm

Done: 53 rows retrieved.

Buttons: Informationen, Erneut installieren, Entfernen

Node.js auf der IBM i - Ausführen



Shell-Skript:

```
PATH=/QOpenSys/pkgsrc/bin:$PATH
export PATH
JAVA_HOME=/QOpenSys/QIBM/ProdData/Java
VM/jdk80/64bit
export JAVA_HOME
cd /home/web_checkin/smyril_checkin_api-
server
node src/server.js
```

CL Programm:

PGM

QSH

```
CMD('/QOpenSys/usr/bin/sh -c +
"/home/web_checkin/smyril_checkin_api-
server/start_api_server.s+
h")
```

ENDPGM

Opt	Subsystem/Job	Aktueller Benutzer	Art	CPU %	Funktion	Status
—	TS11031545	TFU	BCH	0,0	PGM-BKBCFP02	DEQW
—	WEBCHKCLT	MATTHIAS	BCH	0,0	PGM-WEBCHKCLT	TIMW
—	WEBCHKSRV	MATTHIAS	BCH	0,0	PGM-WEBCHKSRV	TIMW

Vielen Dank!

NOVEDAS Software & Systems GmbH

Hermannstraße 18

20095 Hamburg

Mail: office@novedas-sosy.de

Web: novedas.com

novedas-sosy.de (under construction)

Telefon: Telefon: +49 (0)40-180 48 90-60