How to optimize car charging with Camunda 8

incentro

Content
presentation

1. Who are we?
   - Introduction
2. What is going on?
   - Project design
3. What did we do?
   - Process overview
4. How did we implement this?
   - Technical details
5. Demo time
6. Results
7. Questions?

incentro

Who are we?

Today’s speakers

Rick Balfoort
Consultant | Python developer

Maarten van Veelen
Lead Architect

Our figures

For enterprises that want to grow or embrace change: you've come to the right place.

14
Cells

27
Incentro years

500+
Employees

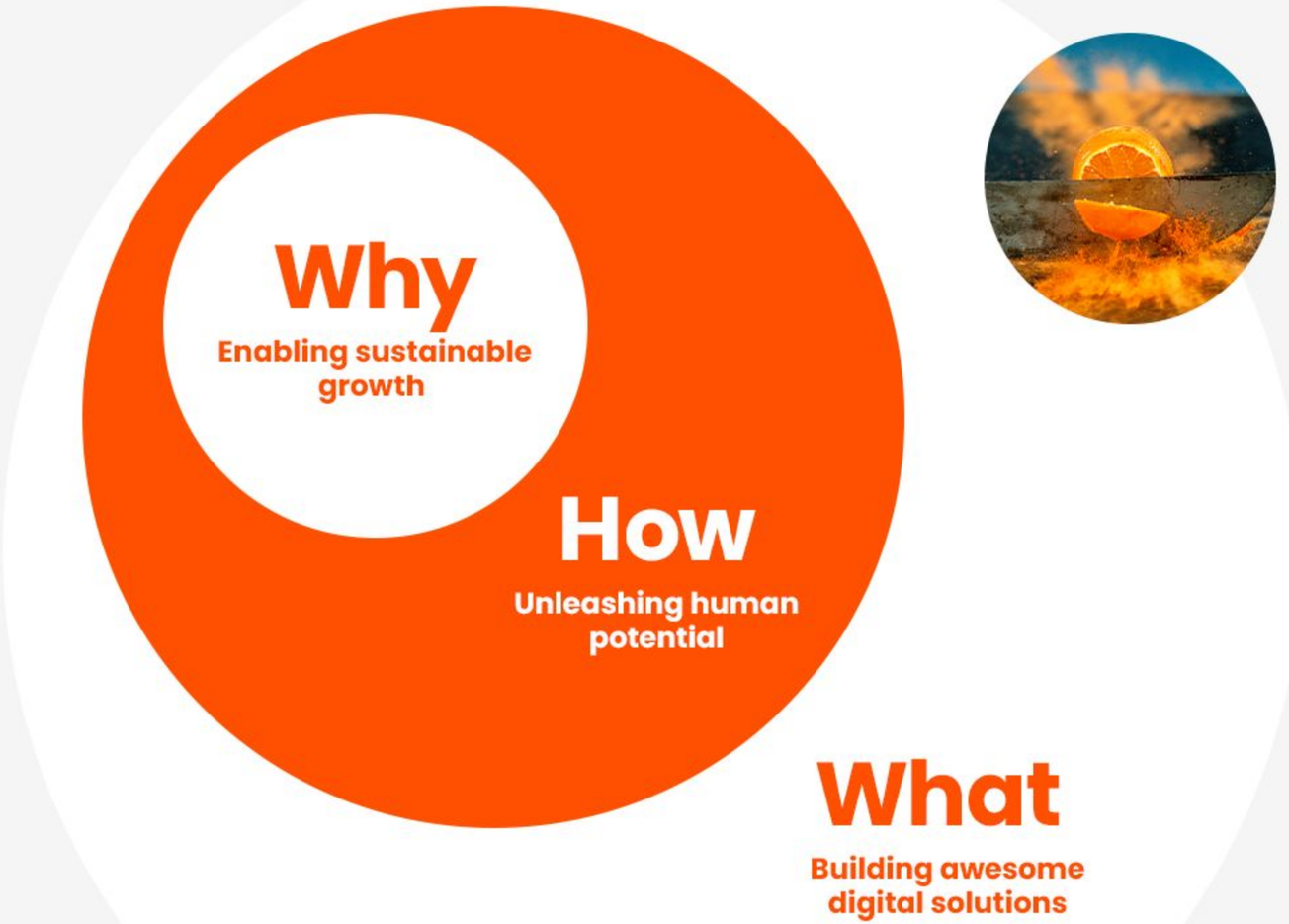
30+
Technology partners

4
Countries

Great Place to Work

Our purpose

Our ambition is to let our clients grow in a sustainable way. Let us grow together?




incentro
What is going on?

Finding a suitable process candidate

How did we get here?

- Worked with platform based on Activiti engine
- All our process questions led to the Camunda forum
- Became enthusiastic about the product
- Became Camunda partner last year with a main focus on C8
- Needed our first process to gain experience with C8

What did we set out to do?

- Automate the car charging process
- See how Camunda 8 would work in combination with Python and diverse endpoints

Charging stations
Business challenges

The challenges

- Emission goal: net 0 in 2025
- 8 charging stations for 40 fully electric cars
- Fully charged cars are not removed
- Empty batteries....

The goals

- Efficient use of the charging stations with the help of Camunda and Slack
- Map the car ‘journey’ using Camunda Optimize for further improvements
- Make the life of our consultants easier

Our parking

incentro

Charging stations
Technical challenges

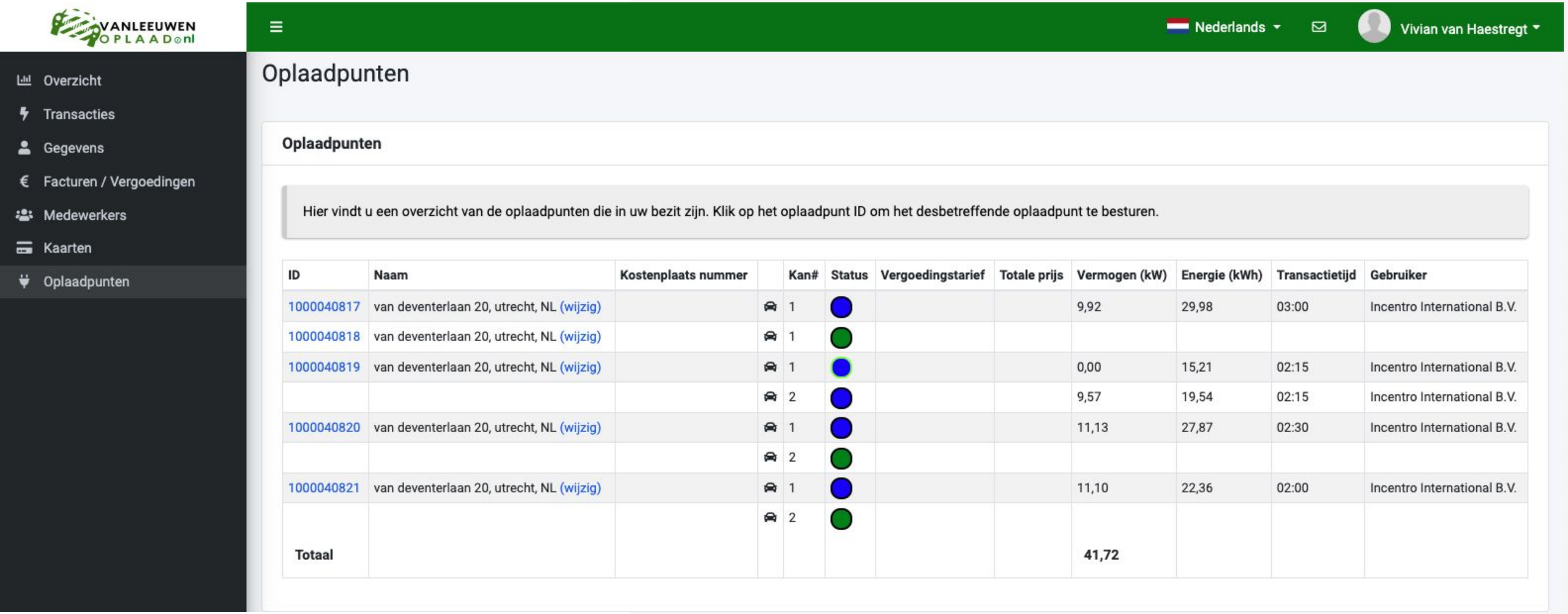
The challenges

- Portal for charging station data but no API available
- Camunda 8 relatively new
- Not much experience with the Python Client

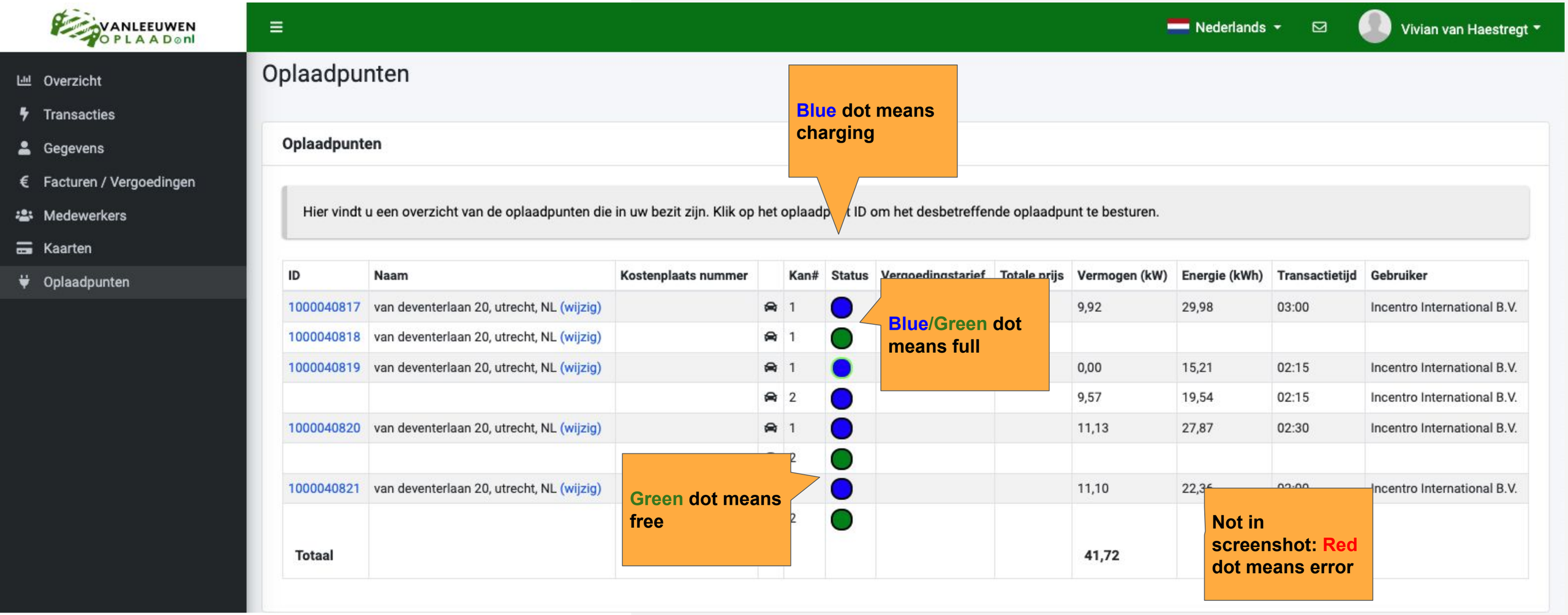
The goals

- Use Camunda to provide insights
- Become proficient with Camunda 8
- Familiarize ourselves with how C8 works with different programming languages and endpoints

What little we had: Charging Stations Admin portal



Connecting the dots…

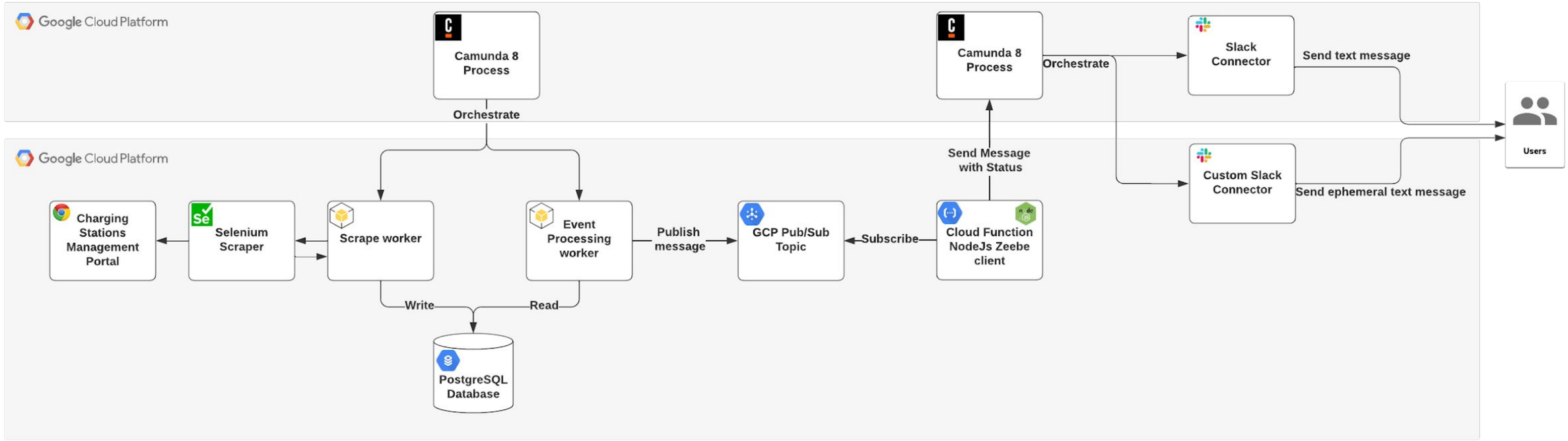


Project design

What we wanted
Solution

What we wanted	Desired solution
Insights into the process of car charging	**Dashboard for managers with information about the process**
Stimulate efficient use	**Give employees insights in how efficient (or not) they use the charging stations**
Make the life of our consultants easier	**Being able to send Slack messages when car is full**
Learn from using Camunda 8	**This project**
Familiarize ourselves with how C8 works with different programming languages and endpoints	**Use as many different techniques as possible**

Design

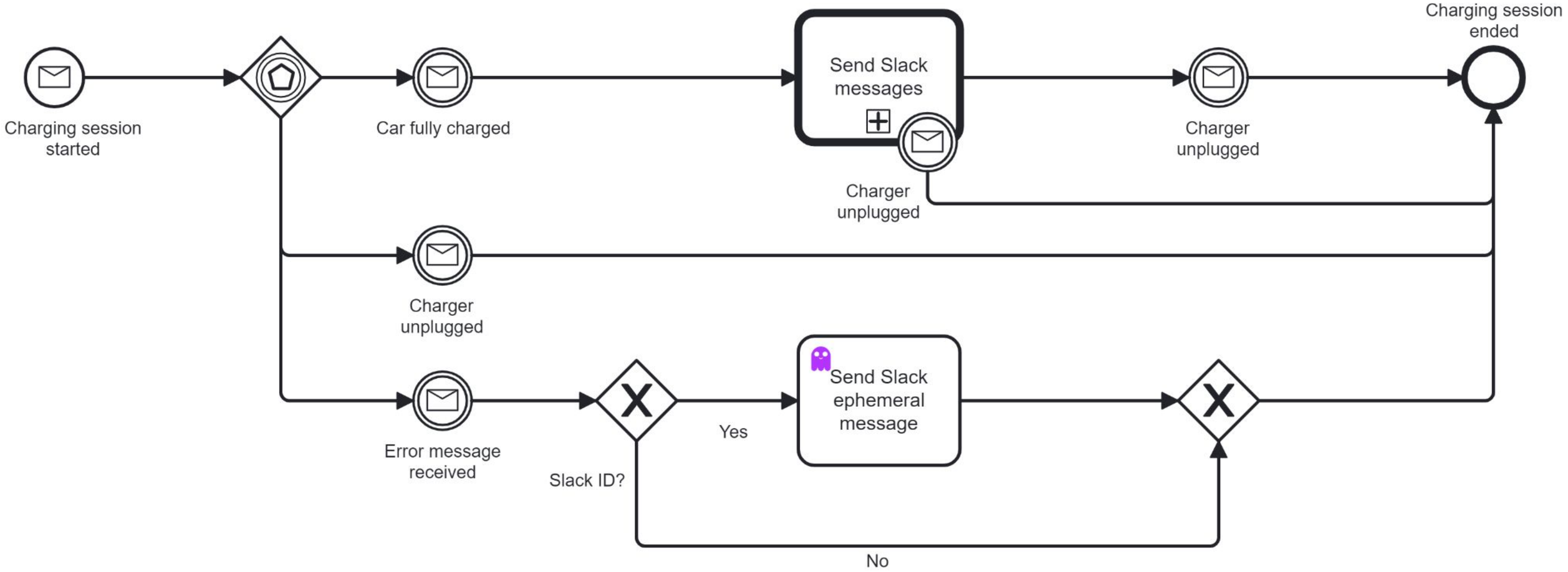

Google Cloud Platform
Camunda 8 Process
Orchestrate
Google Cloud Platform
Charging Stations Management Portal
Selenium Scraper
Scrape worker
Event Processing worker
Publish message
GCP Pub/Sub Topic
Subscribe
Cloud Function NodeJs Zeebe client
Send Message with Status
Write
Read
PostgreSQL Database
Camunda 8 Process
Orchestrate
Slack Connector
Send text message
Custom Slack Connector
Send ephemeral text message
Users

incentro

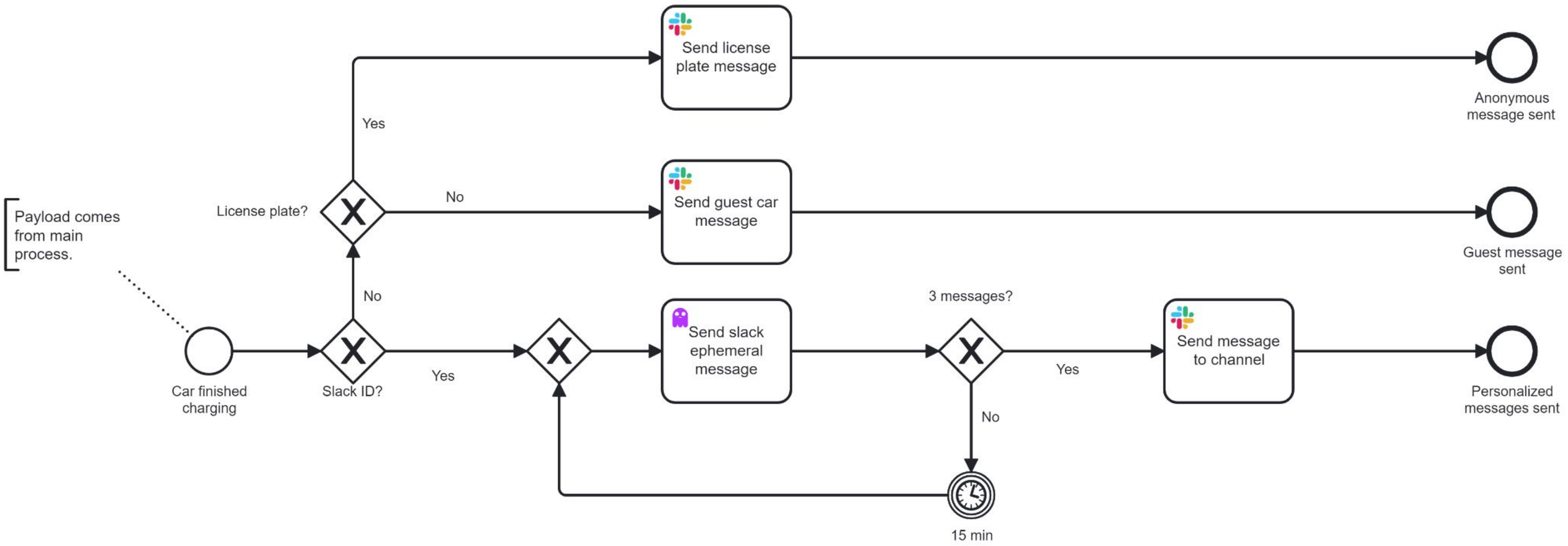
What did we do?

BPMN 1: ScrapeEventsProcess

BPMN 2: IncentroChargingProcess



BPMN 3: SlackSubProcess



Slack messages

01

Personal ephemeral message

Your car is fully charged. Please remove it from the charging station so other colleagues can use it.

02

Group message

Hi R. Balfoort your car is fully charged. Please remove it from the charging station so other colleagues can use it.

03

/availability

Currently there are 3 out of 8 charging stations occupied.


incentro
How did we implement this?

Web Scraping: Python

What is it?

- Python can work with the Selenium library to scrape data from websites

Why did we use it?

- We scrape the data since the vendor portal did not have an API
- Python & Selenium library → because I (Rick) am a Python developer (and fan) and we set out to gain experience with the community client (PyZeebe)

What did we do with it?

- Pyzeebe worker that scrapes charging stations portal
- Returns one of four statuses:
  - Available
  - Charging
  - Charging full
  - Station fault

  → Based on these statuses we deduce events
- Challenging part was to host workers on the VM. There were chromedriver issues which did not occur locally

Data storage: PostgreSQL

What is it?

- A SQL database

Why did we use it?

- In order to decouple scraping and events, we store the data

What did we do with it?

- Data from the vendor portal is refreshed every 15 minutes.
- We compare the data every refresh to deduce events
- The actual charging process is started when a start event is deduced based on the data in the DB

Messaging: Google Pub/Sub

What is it?

- Asynchronous and scalable messaging service on GCP

Why did we use it?

- Our vendor of choice, used to working with it
- Easy to quickly process incoming messages using Google Cloud Functions

What did we do with it?

- Event worker publishes messages to our topic in the cloud
- The cloud function ‘translates’ these messages, using NodeJS
- The ‘translated’ messages are sent to corresponding message events in Camunda

```
function processMessageImpl(body){

    try{
        const bodyJson = JSON.parse(body);

        zbc.publishMessage({
          correlationKey: bodyJson.chargerId,
          name: bodyJson.event,
          variables: {
            chargerId: bodyJson.chargerId,
            card: bodyJson.card,
            name: bodyJson.name,
            email: bodyJson.email,
            slackId: bodyJson.slackId,
            energyTotal : bodyJson.energyTotal,
            transactionTime : bodyJson.transactionTime,
            message_count : bodyJson.message_count,
            kenteken : bodyJson.kenteken
          },
          timeToLive: 600000
        });
```

```
/**
 * Triggered from a message on a Cloud Pub/Sub topic.
 *
 * @param {!Object} event Event payload.
 * @param {!Object} context Metadata for the event.
 */
exports.processMessage = (event, context) => {
  const message = event.data
    ? Buffer.from(event.data, 'base64').toString()
    : '{}';
  processMessageImpl(message);
};
```

Alerts: Slack

What is it?

- Communications platform

Why did we use it?

- Slack connector readily available in C8
- Slack was already widely used within Incentro

What did we do with it?

- We used the out-of-the-box connector to send messages to a slack group chat
- We also send ephemeral messages, for this we built a custom connector
- We use a combination of both messages to alert people when their car is full or when there is an error during charging

Insights: Optimize

What is it?

- Camunda dashboards to give insights into processes

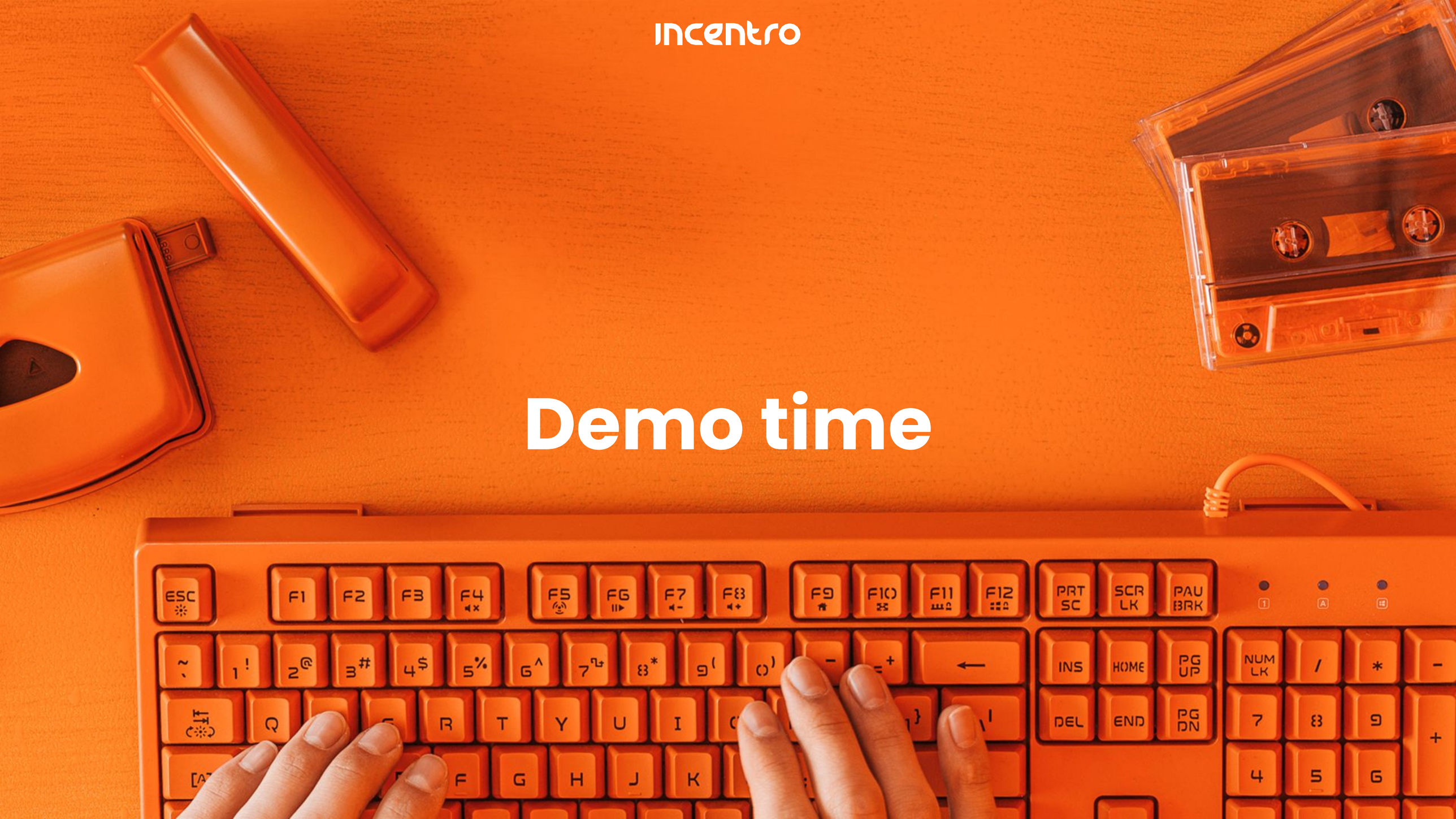
Why did we use it?

- Easy and efficient way to track and improve our processes
- We got a ‘name-and-shame’ dashboard

What did we do with it?

- We created a dashboard with standard and custom reports
- Specifically, we created a ‘name-and-shame’ dashboard which shows which person keeps their fully charged car at a charging station the longest

Demo time



Results

It works!

We are currently running this process internally and it has been a success. We will keep it running and based on feedback will even release more functionalities.

C8

Experience with diverse endpoints and Python

Visibility

of the car journey. The insights we have made charging more transparent and efficient

Efficiency

of the charging process. People do not have to manually send messages and are nudged to remove their cars quicker

incentro

Key Takeaways

- Camunda 8 is a perfectly suitable platform for solving our business and technical challenges
- This use case really helped in explaining our colleagues what Camunda exactly is capable of
- Most work went into the web scraping, new ootb connectors can be of help
- The Python client helped to quickly develop workers and decrease development time


incentro
Questions?