

# Haecksenkarte

**Format:** Projekt

**Status:** aktiv

**Termine:** unregelmäßig, online

**Hüte:** n/a

**RocketChat:** #haecksenkarte

**Mailing Liste:** haecksenkarte@

## About

Die Haecksenkarte ist ein Projekt, das lokale Gruppen, Spaces uvm von und für Haecksen auf einer Karte zeigt.

## HowTo

Unterstützung & Feedback jederzeit gerne per Mail an haecksenkarte@ oder im Rocket #haecksenkarte.

- Projektdoku Haecksenkarte
- technische Doku Haecksenkarte
  - technische Doku Haecksenkarte V2.0
  - Datenmanagement Haecksenkarte
  - Internationalisierung Haecksenkarte
- sonstiges
  - Haecksenkarte Workshop @c38c3

# Projektdoku Haecksenkarte

## Ansprechpersonen

**Projektkoordination:** Nora, lillipulaura

**Technische Umsetzung V2:** zombielypse

## Kommunikation

**Rocket-Chat:** [#haecksenkarte](#)

**E-Mail:** [haecksenkarte@lists.haecksen.org](mailto:haecksenkarte@lists.haecksen.org)

**ML beitreten:** [haecksenkarte-join@lists.haecksen.org](mailto:haecksenkarte-join@lists.haecksen.org)

## Get involved

Wir brauchen euch und euer Feedback!

Wenn ihr Input habt oder euch einbringen wollt, schreibt uns am besten im Rocket-Chat im Channel [#haecksenkarte](#) eine Nachricht.

Wir wollen außerdem Hackathons auf den nächsten Chaos/Haecksen-Events veranstalten, um die Weiterentwicklung voranzutreiben.

## Das Projekt in a nutshell

Am Anfang waren da das CCCamp2023, das gemeinsame Interesse an Geodaten, wenig Ahnung von Web-Programmierung - und eine fixe Idee: Lass uns doch eine Webkarte erstellen, die zeigt, wo Haecksen haecksen! Zwei gute Gründe für dieses Projekt:

1. Die Haecksen-Community ist enorm gewachsen und doch wissen viele gar nicht, wo sie andere Haecksen IRL treffen können. Eine Karte verschafft einen optischen Überblick und hilft bei der Vernetzung.

2. **Bonus:** Wir dürfen mit Geodaten rumdaddeln und lernen ein bisschen programmieren. :-)

# Projektstatus

**We are live!** <https://map.haecksen.org/>

In der Anfangsphase haben wir uns auf die Entwicklung eines Karten-Dummys fokussiert. Dieser enthält zunächst nur ohnehin öffentlich verfügbare Informationen zu bereits existierenden Haecksen-Gruppen.

Die erste (statische) Version der Karte ist seit Anfang Juli live via Subdomain <https://map.haecksen.org> erreichbar. Außerdem ist die Karte auf der Haecksen-Webseite unter <https://www.haecksen.org/lokale-gruppen/> eingebettet.

Im Oktober gab es dann noch ein Update auf V2 mit einigen neuen Features und danach noch viele mehr und natürlich auch immer wieder Bugfixes. Die aktuellen Todos gibt's hier im Pad und Issues auch direkt im Gitlab Repository.

## Diskussion

### Der ursprüngliche Plan und warum wir ((sehr) vorübergehend) davon abgerückt sind

Ursprünglich sah die Idee eine Haecksen-Umfrage vor, an der sich Haecksen freiwillig beteiligen und lokal verorten können. Wir haben uns viele Gedanken über diese "Datenerhebung" gemacht. (Wie) lässt sich so eine "Haecksen-Zählung" datenschutzfreundlich umsetzen? Es sollten keine oder möglichst wenige personenbezogene Daten erhoben werden und trotzdem brauchbare Informationen gesammelt werden, die ein geografisch sinnvolles "Clustering" ermöglichen.

**Dieser Plan ist noch nicht tot, nur aufgeschoben, bis wir eine gute Lösung dafür gefunden haben.**

\* **Anm.:** Für unseren Anwendungsfall muss eine **brauchbare** Karte aus unserer Sicht **nicht präzise** sein. Ob in einer Region beispielsweise nun genau 8 oder 11 Haecksen leben oder nur ungefähr 10, spielt eine untergeordnete Rolle. Wie auch immer die Karte am Ende aussehen wird: Sie erhebt keinen Anspruch auf Vollständig-/Genauigkeit (ausdrücklich verzeichnete Gruppen, die Neu-Mitglieder aufnehmen möchten, sollten natürlich erreichbar sein).

- Anmerkung: momentan sind Haecksen Cluster auf Ebene Landkreis/kreisfreie Städte vorbereitet

# Lösungsideen

- Eine App/ein interaktives Tool entwickeln, auf dem sich Haecksen ohne weiteren Angaben mit einem Klick in einem Landkreis oder einer ähnlich groben geografischen Region verorten können. Neu-Haecksen könnten nach der Anmeldung in der Mailingliste einen Link zu diesem Tool (+ ggf. einen Einmal-Code zum Freischalten) erhalten und ihre Eingabe machen
  - Gegenvorschlag: Klassiker "Tan-Liste" (mit entsprechend komplexen, alphanumerischen Werten, Mindestlänge 8 Zeichen), Begründung: 1 Haeckse sollte deutlich mehr als 1 Eintrag machen können, zum Einen für Picnic Termine, Gruppen usw, aber auch für sich selbst (zB separater Arbeits-/Wohnort oder wechselnde Wohnorte)
  - gültige Haecksen sollten 1 neue Tan Liste anfordern können (eher nicht automatisiert, aber wenn genug zur Verfügung gestellt werden ist das auch eher selten der Fall)
  - außerdem sollte jeder Eintrag 1 Pin erhalten, mit der dieser Eintrag bearbeitet werden kann
  - ohne Bindung an Haecksen-E-Mail-Adresse bleibt dann ggf bei Austritten etwas Datenmüll übrig, aber das dürfte vernachlässigbar sein

## Credits to

- melzai für die Inspiration
- mapc für die technische Umsetzung (Hosting)
- Drakulix für Einrichten des Cronjobs
- Das Adminen-Team, die Berliner Haecksen, mika und zombielypse für das kritische Feedback und Inspiration
- naerrin für das Einbinden auf der Webseite

# technische Doku

## Haecksenkarte

# technische Doku

# Haecksenkarte V2.0

## Hosting

- statisch auf [map.haecksen.org](https://map.haecksen.org)
  - gesourct im iFrame auf <https://www.haecksen.org/lokale-gruppen/>
- Content wird per Cronjob von Gitlab geholt (maintained by Drakulix)
- ssh Zugänge haben: TODO
- Server: TODO

## Code

- Repository auf GitLab:
  - Gruppe: <https://gitlab.com/haeckmap>
  - HaeckMap: [https://gitlab.com/haeckmap/HaeckMap\\_v2.0](https://gitlab.com/haeckmap/HaeckMap_v2.0)
  - Tickets: <https://gitlab.com/groups/haeckmap/-/issues>

## Design

- js only, weil statische Webseite
  - Frameworks für Map:
    - Leaflet (Upgrade auf V1.9.4 für Koordinaten in Karte auswählen)
      - TODO: ggf bessere Version finden, diese scheint etwas unperformant zu sein
    - Topojson (für LK Layer)
    - geolib.js (zur exakten Bestimmung des Landkreises von einem Paar Koordinaten)
    - leaflet-fullscreen (Captain obvious)
    - ansonsten (bis jetzt) nur pure Vanilla js und css
- intensive Verwendung von template literals zum HTML bauen -> das ist hier so sinnvoll, weil perspektivisch auf eine App migriert werden soll und template literals sich leichter in echte Templates migrieren lassen

# Repository Struktur

- css: captain obvious
- data:
  - external (Daten aus externen Quellen)
    - topojson (Landkreise mit Außengrenzen)
  - generated (Daten, die in der App in der DB stehen)
    - counties (Länder, Bundesstaaten und Kreise mit Schwerpunkten)
    - group\_data
    - translations
  - field\_mapping: Mapping von Daten auf UI
  - constants: Captain obvious
- dist: Platzhalter für minified js/css
- img: captain obvious
- js:
  - main: (View-übergreifende) Funktionen getriggert durch User und View-übergreifende Funktionen
    - functions: zB Sprachen wechseln, user-defined styles finden, subscribe RSS
    - listener: alle Funktionen, die an Buttons usw hängen
  - model: Klassen für Datenstrukturen und parsen der Daten
    - county\_map: Modell der Landflächen, benötigt zum Bestimmen eines Standorts anhand von Koordinaten
    - local\_group: Modell eines Eintrags in den Gruppen/Space-Daten
    - parse\_data: Logik zum Einlesen/Konvertieren der Daten
  - template: Funktionen, die das HTML generieren (template literals)
    - hier gibt es für jede Fläche in der UI 1 file, und innerhalb deren 1-mehrere Funktionen, die Teile dieser UI erzeugen
  - views: Aufbau und (View-spezifische) Funktionen für Views
    - hier gibt es auch für jede Fläche in der UI 1 file, und darin sind die Funktionen dieser UI-Bestandteile hinterlegt, zB die Suchfunktion in der Liste
  - main.js: Script, das ganz am Ende geladen wird und nur die Initialisierung der Verarbeitung auslöst
- dev: Templates/snippets aus denen die Website in der Pipeline gebaut wird, und die externen js Importe für den offline Gebrauch Dev/Test
- generate\_files.py: generiert in der Pipeline die index(\_xx).html files, die RSS feed files, und die js Dateien mit den generierten Daten
- download\_newest\_artifacts.sh: Skript zum Download des fertigen package (zur Verwendung im cron Job und für Testzwecke)
- html/index\_dev(\_offline).html: HTML file für Dev/Test Zwecke

## Details

- data/generated
  - das bedeutet, dass diese Daten aus einer "internen" im Sinne von Haecksen-eigenen Quelle kommen, aber nicht manuell erzeugt/gepflegt werden

- der Einfachheit halber können bis V3 auch manuelle Änderungen vorgenommen werden, die müssen dann nur ebenfalls in der DB eingepflegt werden
- data/generated/counties
  - Diese enthalten nur den Schwerpunkt einer Fläche, keine Kette von Koordinaten, mit denen man eine Grenze in die Karte zeichnen könnte. Das ist sozusagen eine lightweight-Variante, um eine Detektierung einer Fläche zu einem Paar Koordinaten feststellen zu können. Ohne diese zusätzlichen Daten, würde der Algorithmus zur Bestimmung der Daten zu Land/Bundesstaat/Landkreis nur die nächstgelegene Fläche ausgeben, auch wenn diese fernab der gesuchten Koordinaten liegt. Demo: einfach mal in der fullFeatured Version über Amerika oder Afrika Add new anklicken, dann werden die Felder vermutlich mit Island bzw Zypern oder so ausgefüllt (Europa ist halbwegs abgedeckt)
- field\_mapping
  - im field\_mapping werden die Input-Daten mit Attributen zur internen Verarbeitung assoziiert, zB:
    - welcher Datentyp sie sind, damit im Formular das richtige Element dargestellt wird (zB DropDown, Radio Buttons, Checkboxes oder Text Input)
    - für alle Datentypen mit fixen Optionen stehen hier auch die gültigen Optionen (wenn die nirgendwo stünden, könnte man sie ja auch im Formular nicht zum Anklicken anbieten)
    - für welche "types" sie gültig sind (zB ist das Picnic Date nur für Picnics gültig)
    - wo sie angezeigt werden sollen (Anzeige in Map/Liste ja/nein, Anzeige im Formular ja/nein)
    - mandatory: Pflichtfeld ja/nein, editable: editierbar ja/nein, calculated: Wert wird programmatisch bestimmt (zB "Standort", das aus den Einzelangaben zusammengesetzt ist)
    - privacy: das Feld kann im Formular als privat markiert werden, dann wird in der Mail unter metadata -> privacy dieses Feld mit aufgelistet, das Import-Tool berücksichtigt diese Angabe dann und setzt das Flag private in der DB auf true - diese Daten werden dann nicht exportiert / von V3 nicht angezeigt
      - Bitte beachten: momentan ist dieses Flag für alle Einträge deaktiviert
  - das field\_mapping ist die wichtigste Stellschraube zur Anpassung der UI - hier werden u.a. die Reihenfolge der Darstellung der Daten für die UI, ihre Gruppierung, ein-/ausblenden, Eigenschaften von Feldern usw gesteuert
  - die allermeisten Anpassungen an der Darstellung der Attribute lassen sich allein damit lösen
- js/model/parse\_data
  - da die Daten jetzt nicht nur 1x in der Map gebraucht werden sondern auch 1x in der List view und 1x im Formular Änderung melden, werden sie nicht mehrmals gelesen sondern einmal am Anfang und als Objekte in eine Liste gespeichert, siehe local\_group...
- js/model/local\_group
  - ...diese Objekte ("LocalGroup"s) haben alle wichtige Funktionalität direkt an sich, wie zB das Aggregieren der Addressdaten, das Ersetzen der Links, das Erzeugen eines json snippets von sich selbst, das Bestimmen von Koordinaten/Standortdaten



passend zueinander und natürlich alle Daten (“Attribute”)

- local\_group “harmonisiert” die Daten von in- und output mit der UI - ein Record, der von local\_group exportiert wird kann exakt so auch importiert werden und würde immer exakt gleich in der UI aussehen
- das hilft primär beim Erzeugen des outputs für die E-Mail, hat aber zB auch das Gimmick “Preview als Draft in der Map” abgeworfen

# Daten

- momentan als json gespeichert, Format:

```
{
  "type": "Feature",
  "properties": {
    "fid": 1,
    "typ": "space",
    "count_fictive": 18,
    "space_name": "name"
    // more attributes
  },
  "geometry": {
    "type": "Point",
    "coordinates": [
      9.0,
      52.0,
    ],
  }
}
```

# Datenmanagement

## Haecksenkarte

### Daten

- momentan als json gespeichert, Format:

```
{
  "type": "Feature",
  "properties": {
    "fid": 1,
    "typ": "space",
    "count_fictive": 18,
    "space_name": "name"
    // more attributes
  },
  "geometry": {
    "type": "Point",
    "coordinates": [
      9.0,
      52.0,
    ],
  }
}
```

- Repository HaeckMapData (private): <https://gitlab.com/haeckmap/HaeckMapData>
- neue Tabellen:
  - create table spaces (id number, type text, location text, x number, y number);
  - create table properties (id number, private boolean, type text, key text, value text);
  - create table options (key text, option text);
- dadurch können beliebige neue Eigenschaften hinzugefügt werden, unterstützt werden dafür auch praktische alle Datentypen, die mir gerade einfallen würden (number int/float, alle strings, boolean, enums -> multiple choice)

- aktuelle Änderungen betreffen vor allem die “Pflichtfelder” (bislang: id, location) die nicht mit Haecksen Clustern vereinbar sind und wenn wir ehrlich sind auch so auf der Karte Probleme machen (Einträge ohne Koordinaten, problematische Namen in der Liste bei exotischen Ortsangaben wie “Frankfurt/Mainz/Wiesbaden”)
  - neue Pflichtfelder: id (autogenerated als neuer Eintrag wenn kleiner 0), type [space,group,individual,picnic], name (im Sinne eines Bezeichners), latitude und longitude (letztere für Haecksen individuals von County übernommen), email (als minimum requirement zur Kontaktaufnahme, fallback info@)
  - optional ist alles andere und alles andere kann theoretisch auch als privat markiert werden (außer boolean und arrays / multiple choice siehe Feature #25)
- außerdem gibts ne neue Tabelle für die County Daten, damit aus den gespeicherten Haecksen die Haecksen Cluster aggregiert werden können (Cluster werden nicht als Entitäten gespeichert)
- wenn irgendwann in der Zukunft flask oder fastapi den Zuschlag fürs Backend bekommen, dann kann der Code größtenteils recycelt werden

# Internationalisierung Haecksenkarte

## Internationalisierung

Die Internationalisierung der Haecksenkarte erfolgt über ein json, in dem alle Texte und Label in verschiedenen (beliebigen) Sprachen hinterlegt werden können.

Wenn ihr was übersetzen möchtet, dann braucht ihr zuerst das [translations.json](#) aus dem Repository. Öffnet die Datei am besten in einem Editor, der json supportet. Json ist ein Format, in dem Key-Value-Paare strukturiert gespeichert werden können. Die Datenstruktur sieht so aus:

```
{
  "de": {
    "label_name": "Label Text",
    "button_name": "Text auf dem Button"
  },
  "en": {
    "label_name": "translated label text",
    "button_name": "translated text on the button"
  }
}
```

Beim Bearbeiten muss die richtige Stuktur beibehalten werden, d.h. insbesondere Kommas und Anführungszeichen müssen genauso verwendet werden. Ein guter Editor sollte euch zeigen, wenn beim Editieren etwas kaputt gegangen ist.

Bitte gerne auch im Rocket [#Lost-In-Translation](#) vorbeischaun :)

## Status

Sprache	Status
de	live
en	live
es	in progress, zombie
pt	in progress, zombie
fr	live, headlina, phoibi

Sprache	Status
pl	todo
fi	todo
da	todo
nl	todo
ar	todo
ru	todo
ukr	todo
it	todo
tr	todo

sonstiges

# Haecksenkarte Workshop

## @c38c3

### Part 1 - 30 Min

- Präsi der Karte über Lauras oder Noras Laptop
- Wie ist es zum jetzigen Stand gekommen?
- Selbst ausprobieren > ggf. Mail versenden
- Erstes Feedback einholen

### Part 2 - 30 Min

2a: Fehlende Daten, Wissenslücken vorhanden über lokale Haecksen-Aktivitäten

- Grundproblematik: Ansprechbarkeit einzelner Personen vs. Datenschutz
- Wäre Anzeige Nickname ok?
- Ziel: Netzwerkwerkeffekt
- Derzeitige Mindestangaben: mindestens 1 aus E-Mail, Mailing-Liste, Webseite
- Was gibt es in Sachen Datenschutz zu beachten?

2b:

- Welchen Zweck soll die Karte zukünftig erfüllen?
- Stimmungslage erfassen
- Welches Konzept wird präferiert?
- Beachten, dass die Karte öffentlich erreichbar ist

### Part 3: 30 Min

- Accessibility der Webseite
- Optisches
- Technisches
- Features
- Sonstiges

# Feedback

(Progress Stand 29.01.2025, Updates werden nur im Pad eingetragen)

☒ (~~Leaflet-Plugin~~) Clustern ist eingeplant

☐ (Daten) Mischform mit Gruppe und Space, wenn es einerseits einen festen Ort und daneben andere Events

- Einzig bekannte Fall, auf den das zutrifft, wird gerade intern geklärt > München
- keine programmatische Anpassung nötig
  - weitere Unklarheit, die aufgetreten ist: Karlsruhe. es ist auch einfach inkonsistent und nicht logisch und auch beim review/ux test schon negativ aufgefallen, sollte mE schon richtig umgesetzt werden
  - eure Stichprobe war da btw auch einfach zu klein, Schweiz hat sich zB auch als Gruppen angelegt und trifft sich in spaces. es ist einfach unsauber Gruppen zu Spaces zu erklären, nur weil sie sich in Spaces treffen
  - plus: Gruppen, die sich ne "feste" Adresse anhängen kommen zu 100% nicht auf die seltsame Idee, sich selbst zum Space zu deklarieren (siehe zB CH, NBG) und wenn Gruppen diese Eingabefelder erst gar nicht hätten, dann hätten wir in der Konsequenz wohl eher keine Daten als dass da jemand auf die Idee käme den Typ dafür zu ändern
- Einigung: wird geändert, spaces werden haeckspaces, gruppen können sich in n spaces treffen, spaces können n gruppen beheimaten, zombie machts

☒ ~~obsolet mit Sidebar (UI, Internationalisierung )~~ es fehlt ein Hinweis auf die Neuland-Option

- Mögliche Lösung > Anleitung erweitern
  - nicht hierfür, aber hab mir tatsächlich auch so ne kleine grafische "tour" durch die map als todo aufgeschrieben, braucht aber css und screenshots mit Bearbeitung, steht auf meiner Liste also ziemlich weit unten weil mühselig
- Kurzer Erklärtext zu Beginn des Formulars, bspw. unter "Anleitung": "Digitale Treffen können dem virtuellen Land "Neuland" zugeordnet werden, siehe unten im Auswahlmenü "Standortangaben > Land"."
- unnötiger Aufwand, weil ~~obsolet mit Sidebar~~

☒ ~~obsolet mit Sidebar (Daten, Sidebar)~~ Onlineevents eigene Farbe und als extra Liste anzeigen > Cluster

- Siehe vorheriger Punkt
  - das war schon das Ergebnis davon, wenn man digitales pauschal nur noch in der Sidebar hätte, dann ist es per se nicht mehr als Marker auf der map sondern in einem ☐ button als "cluster" zusammengefasst, also keine extra Farbe nötig, nix dergleichen

☒ ~~obsolet mit Sidebar (UI, Daten/Internationalisierung)~~ Wenn "nur virtuelle Treffen" ausgewählt, dann Rest ausgrauen oder Hinweis, dass man nicht glaubt mehr setzen zu müssen

- Geringe Prio. Wenn Anleitung optimiert ist, wsl. nicht mehr nötig, weil Formular übersichtlicher wird.



- Formular hatte ich mir überlegt sollte für digitale Themen (nicht nur Events) ein URL Feld als "Adresse" bekommen, das ist dann das einzige Feld in Standort
- ansonsten same s.o., sowieso obsolet mit Sidebar

☒ (UI, fixed) bei neuland keine bundesländer mehr listen

☐ (Daten, Sidebar) Wie soll Neuland behandelt werden bzgl. Koordinate

- In der Kartenansicht Sidebar nur für Neuland einfügen
  - wieso?
  - nö, doch für alle, Popup nur mit Name, Standort + Details Link, Sidebar Detail Features mit Mika
- Platzhalter-Koordinate in 0/0
  - braucht es dann nicht mehr
- Bei virtuellen Veranstaltungen transparente Farbgebung
  - wieso denn das? in der Karte? das wäre doch auch obsolet mit Sidebar

☐ (UI, Internationalisierung) Email-Hinweis: Das ist die Mail, die in der Karte als Kontaktmöglichkeit erscheint.

- Feldspezifischer Kommentar bzw. ein Infobutton, der bei Mouse-over Erläuterung anzeigt
  - die hints und placeholder können einfach angepasst werden, kleine popups sind erheblich schwieriger und nicht einfach barrierefrei zu machen
  - "Alle Infos, die nicht auf der Karte erscheinen sollen, bitte Häkchen auf "privat" setzen. Info wird dann nur vom Team Haecksenkarte zur Kontaktaufnahme verwendet."

☐ (Daten, UI) Anderes Listing für Länder, Sortierung: D-A-CH-Länder sowie Neuland nach oben, optische Trennung zu allen anderen Ländern

- Neuland entfällt mit Sidebar
- Dach nach oben (so war es lange) wirft Probleme auf: dann können die Einträge dahinter nicht mehr barrierefrei angesteuert werden, einzige mögliche Lösung: ein benutzerdefiniertes Element was jemand programmieren müsste, der CSS und js kann, mindestens jedoch css

☐ (App) X für Formular schließen fehlt (mobile)

- Geringe Prio
  - (es gibt btw auch einfach kein "Formular schließen", es kann keins geben) dürfte mit Anleitung auch geklärt werden, war mit an Sicherheit grenzender Wahrscheinlichkeit ein Bedienfehler, weil das single Page Konzept, naja, kein schönes ist, ein Burger menu würde das Problem auch erheblich improven, also falls wir jemanden mit CSS Know-how auftreiben können

☐ (TBC, bei iphone test nachgehakt) Wenn Punkt ausgewählt und Klick auf Report update / Event ändern auf iPhone: Das darauf folgende Formular ist nicht vorausgefüllt

☐ Nora: Fehler reproduzieren

☐ s.o. wahrscheinlich Bedienfehler aufgrund des single Page problems

☐ (App) Klickbare Karte regulär anbieten > Backend, um IDs an Einzelpersonen bzw. Emailadresse zu knüpfen

- Prüfprozess nötig, bpsw. über eine tanliste oder pin

- Verknüpfung mit Mailingliste möglich oder existierendem Onboardingprozess?
  - quick & dirty: ne Liste mit Pins in nem geschlossenen Bereich zur Verfügung stellen, zur SB
- ☐ (App) Ausbau Backend erfordert eine Person, die uns mit Serverhosting hilft
  - web app geht nur dynamisch und das geht bei den haecksen nur mit docker
  - für server config brauchen wir dann sonst niemanden

Lexie (Regensburg), dorami (münchen), ulli

- Im Formular “neu” der Hinweis hinter Land: “(automatisch)” ist irreführend, besser entfernen
  - indeed, ~~so gut wie~~ gefixt