Abstimmungsdokument der Bachelorarbeiten von Kai und Sebastian

Sebastian Repo: https://gitlab.uni-kassel.de/K066A/kgeo

Living Document => wird von Kai, Oliver, Sebastian und Bernd zum Zwecke der Abstimmung und Präzisierung nach Bedarf angepasst und erweitert. Bitte jeder in seiner Farbe texten. Am Anfang bitte die Abgrenzung/Definition was der Inhalt der Bachelorarbeit sein soll bzw. alles weitere danach auf den Folgeseiten.

1) Abgrenzung und Definitionen zum inhaltlichen Rahmen der Bachelorarbeiten.

Generell:

- Sebastian hat den Schwerpunkt der Codierung, Kai den Schwerpunkt der Visualisierung.
- Sebastian sorgt für a) das mirroring relevanter Files mit Basisdaten bzw. b) die Konvertierung in ein JSON Format inkl. Adaptierungen bzw. Filtering bzw. c) die Konsolidierung und Anreicherung dieser Daten in neue, zweckmäßige Dateien.
- Kai sorgt dafür, dass interessante und relevante Daten aus der Arbeit von Sebastian, aber auch von weiteren Quellen (PeeringDB) auf openstreetmap gemappt werden. Dies ist für beide Arbeiten relevant, da damit auch die Ergebnisse von Sebastian visuell dargestellt werden können.
- Beide Arbeiten sind eine wichtige Grundlage für weitere Arbeiten in den kommenden Jahren. Bitte daher auf Präzision und Dokumentation achten und daran denken, dass andere den entstehenden Code potenziell ändern, erweitern und fortführen werden.
- Unsere “API” ist in diesem Projekt per Definition das JSON Format. Daten werden so in ein JSON runtergeschrieben, dass es für den Folgeprozess möglichst direkt als Input verwendet werden kann.
- Alle Codeteile bitte sauber als für sich allein stehend, gekapselt und simpel ausführen. Keine komplexe Kombination von mehreren Aufgaben in einem Code.
- Der großteil der Daten wird von den Quellorganisationen tagaktuell gehalten (Registrydaten, PeeringDB Archiv von CAIDA), andere haben z.B. zweiwöchige Aktualisierungen (Maxmind), andere evtl. auch seltener. Teil der Arbeit ist daher, dass a) ein Plan vorliegt zu welcher Uhrzeit per Cron-Jobs täglich welche Files von wo geholt werden können b) die Verarbeitung dann automatisiert gestartet wird und c) von allem tagaktuell ein “latest” File lokal vorliegt und das File vom Vortag archiviert wird. (Frage: wer macht hiervon was?, wie kurz und rudimentär können wir den Teil halten, um nicht zu viel Zeit hierfür aufzuwenden?)

Sebastian:
Titel der Arbeit: < bitte hier nochmal in DE/EN einfügen >

Tips für den theoretischen Teil, der aber auch für das Datenverständnis für uns alle relevant ist: (Anm.: ich (Bernd) kann hierbei gerne inhaltlich helfen)

- Wer sind die IANA bzw. die 5 RIR und was machen diese.
- Wie gelangen die 3 fokussierten Ressourcen: ipv4, ipv6, ASN von der IANA über die RIRs, über die LIRs zu den finalen Nutzern?
- Was ist der Unterschied zwischen allocation/allocated und assignment/assigned bzw. Hinweis des Einflusses in welchen Datenfiles was hiervon zu finden ist und was nicht.
- Wieso schauen wir primär auf die LIR Allocations bzw. nur ergänzend auf Assignments (Abgrenzung der soll-Daten und nicht-Daten)
- Beschreibung der Thematik, dass IPv4 in zwei Darstellungen vorliegt: a) Startadresse mit Länge (z.B. in allen RIPE allocations) bzw. b) CIDR (ipv4 im Routingbereich und bei Maxmind Daten bzw. bei IPv6 generell bzw. bei der ipv4 Zuweisung der IANA an die RIRs) - festhalten,

dass wir nicht auf CIDR wechseln können, da sich die RIR allocations nicht an schematisch korrekte boundaries halten die es für CIDR aber bräuchte.

Prozessteil Daten-Mirroring:

- siehe oben - finde die richtigen Quellfiles die für unsere Zwecke relevant sind und stelle durch Quervergleiche fest, was wo drinnen ist, ob es dann für uns komplett ist bzw. ob wir mit dem nro Aggregat arbeiten können. Als Annahme gilt, dass es im Internet im Routing nichts geben darf, was geroutet, aber in diesen Daten nicht als allocated oder assigned zu finden ist. (Anmerkung: geben wird es mutmasslich sicher hijacked Routes von nicht allozierten IP Spaces - das ist aber in dem Projekt zu ignorieren und nur insofern zu checken ob es nicht auf legal allozierten Space hinweist, wo evtl. irgendeine Quelle übersehen wurde)
- Erforderlicher Präzisionsgrad: hoch, da dies eine Trägerbasis für sehr viele weitere Arbeiten und Analysen sein wird. Bitte hier genügend Zeit investieren um 100% zu erreichen und auch checken ob die von IANA historisch direkt assigned prefixes - vor allem die großen /8 auch enthalten sind.

Prozessteil Daten-Konvertierung:

- aus dem oder den festgestellt passenden und lokal gemirrorten Masterfiles, bitte folgenden Split erzeugen: a) ipv4 (Name hier im Dokument: “ipv4-registered-sebastian”), b) ipv6 (“ipv6-registered-sebastian” und c) ASN (“asn-registered-sebastian”) (alles: JSON Format und täglich aktualisiert verfügbar halten als xyz-latest)
  Anmerkung: ipv4 bleibt in Form: Startadresse mit Längeninfo, ipv6 in CIDR Form
- finde das master-file mit ASN zu as-name, füge es zum mirroring hinzu und sorge für ein tagesaktuelles as-names-latest in JSON Format (“as-names-sebastian”)
- aus dem lokalen tagesaktuellen mirror der Maxmind Daten: wir brauchen derzeit nur den Prefix wie vorliegend in CIDR Form und das Land (DE, AT) - d.h.: Vorschlag zur Vereinfachung (?): aus zwei Files mach eines (den country-code foreign-key direkt ersetzen durch das Länderkürzel) - der GPS Code ist wertlos weil die free version nur auf ein Land zeigt, damit ist auch die Accuracy wertlos und auch die Felder wie Proxy-Info etc. brauchen wir aktuell gar nicht => d.h. Ergebnis deines ipv4-geolocation-maxmind-latest in JSON format ist die ipv4 prefixliste in CIDR Form mit Länderkürzel - detto mit IPv6 (ergibt: “ipv4-geolocation-maxmind-sebastian” und “ipv6-geloaction-maxmind-sebastian”)
  Anmerkung: ipv4 und ipv6 liegen dann in CIDR Notation vor.

Prozessteil Routingtabelle:

- ist von Kai zu Sebastian gewandert - evtl. kann Sebastian die Überlegungen von Kai übernehmen? Es braucht eine auszuwählende Quell-Full-Table. Wir reduzieren hier vereinfachend auf eine vollständige. Später passiert hier dann potenziell sehr viel mehr.
  Von der ipv4 und ipv6 table soll ein JSON entstehen, das zum Inhalt hat: ipv4 bzw. ipv6 Prefix in CIDR Notation, sowie der vollständige AS-Pfad sowie als eigenes Feld: das letzte ASN im AS-Pfad welches das Originate ist => d.h. bitte das pro Prefix extrahieren und als drittes Datenfeld abspeichern (weil: wir brauchen sowohl Pfad als auch Originate-Info). Später kommen dann Infos wie MED, LOCALPREF etc. mit hinzu - aktuell kann das unterbleiben. (Ergebnis: “ipv4-fulltable-sebastian” und “ipv6-fulltable-sebastian”)

Prozessteil: ASN anreichern

- Bitte ein neues JSON “asn-extended-sebastian” mit dem asn-registered-sebastian und anreichern mit dem ASN-Namen aus as-names-sebastian und zwei JA/NEIN Feldes namens “in-global-ipv4-routing-table” und “in-global-ipv6-routing-table - Das JA/NEIN ist eine Suche nach dem ASN in der ipv4-fulltable-sebastian im Feld AS-Pfad (achtung! nicht im originate!) bzw. ipv6: detto => ergibt eine neue Datei mit ASN, Name, Registrierungsland und der Info ob die ASN irgendwo in ipv4 oder ipv6 vorkommt.

- Bitte ein neues JSON File oder eine Funktion: Zähle die Summe aller ASN pro Registrierungsland => sollte das Pendant zu dem hier sein: https://bgp.he.net/report/world
- Bitte ein neues JSON File oder eine Funktion: Zähle die Summe aller ASN mit einer OR Verknüfung aus in-global-ipv4-routing-table und in-global-ipv6-routing-table => d.h. das hier verkleinert die Anzahl der pro Land registrierten ASN auf die Summe der aktiv/lebenden ASN => das müsste das Pendant zur sichtbaren Anzahl (rowcount) von z.B. dem hier sein: https://bgp.he.net/country/AE oder dem hier: https://radar.cloudflare.com/routing/ae

Prozessteil: Prefixe verorten

- Nimm die Tabelle ipv4-fulltable-sebastian bzw. ipv6: detto und erzeuge ein neues v4 und v6 JSON File mit den Feldern prefix, originate,
  füge ein Feld geolocation-country-maxmind hinzu
  suche in der ipv4-geolocation-maxmind-sebastian (und v6) nach dem prefix oder einem größeren Prefix (größeres Supernet) und übernimm den Ländercode daraus (DE, AT, AE, …)
  Wichtig: wir wissen noch nicht ob und wie viele Prefixe gar nicht in Maxmind existieren, wie viele Prefixe vielleicht doppelt und sich widersprechend vorkommen, wie viele Prefixe eventuell überschneiden widersprüche bilden => hier bitte mal ein paar Testläufe machen und bitte Besprechung dazu was wir daraus lernen, eventuell klemmen wir einen Statuscode zu Maxmind hinzu um festzulegen in welcher Qualität wir das Land gefunden haben? das korreliert mit Punkt ii von Oliver und der Aufgabe eine Ratio zu ziehen.
  Idealfall: wir haben den prefix mit dem originate-ASN und dem Landeskürzel

von Oliver:

> “ii. Ratio bilden: wieviel % der Prefixe haben die gleiche Registration Location wie die Prefix Geolocation?”
> hier müsste man sich noch entscheiden was die Registrierungsreferenz genau ist => discuss …

jetzt kommen wir auch “schon” zum Ziel:

- Bitte ein neues JSON File oder eine Funktion: Zähle die Summe aller unique ASN pro Land. => das ergibt dann das eigentliche Ziel: nämlich der Feststellung, wie viele ASN man pro Land (z.B. UAE) findet, wo Maxmind meint, zumindest einen Prefix dort zu sehen. Das müsste erwartet größer sein als die Anzahl der registrierten und aktiven ASN des Landes.
- Dieser Count für alle globalen Länder dann bitte in eine Table einer Landesliste mit den 3 verschiedenen Counts zusammenfügen (registrierte ASN, registrierte und aktive ASN, laut Maxmind im Land laufende ASN)

Kai:

Titel der Arbeit: < bitte hier nochmal in DE/EN einfügen >

Fragen:

- Wo ist ein AS präsent? Auswahl: ein ASN. Hier die verschiedenen Sichten nutzen: 1) Land der Registrierung, 2) an welchen Peering facilities (aus PeeringDB) präsent, 3) welche Prefixe werden an diesen Facilities announced, 4) geolocation der announcten Prefixe (Maxmind, IP2Loc).
- Wo ist ein Prefix präsent? Auswahl: ein Prefix. Welche Informationen sind für einen Prefix bekannt?

Beide Aspekte im Backend berechnen und dann auf der OpenStreemap Karte visualisieren (mit verschiedenen Layern).

In der Visualisierung waeren das dann Auswahlboxen: 1) Liste von ASNs 2) Liste von Präfixen (oder besser Eingabefeld). Und dann map.html?asn=173 -> zeigt alles ueber AS 173 an .. map?prefix=127.0.0/8 …

- Wer ist in welchem Land

Im Detail:

Theorieteil Notizen:
Softwareengineeringteil beschreiben
Statistik durch Plots aufarbeiten und durcherklären - wie viele ASN pro Land - auch in Verbindung mit grafischen Kartenplots -

1)
schon besprochen und in Arbeit: peeringdb facilities auf eine openstreetmap

2)
idee: alle ixp auf die openstreetmap mappen - ähnlich der https://www.internetexchangemap.com/ - evtl. mit facilities in parallel?
auch interessant: Filtermöglichkeiten auf einer Karte:

- zeige z.B. alle DE-CIX und AMS-IX und LINX und Equinix Standorte auf einer Karte (3 verschiedene Farben)
- zeige die IXPs mit einer Memberanzahl auf der Karte
- zeige nur IXPs ab n Member auf der Karte (n=10, 50, 100 etc…)

3)
aus den daten von sebastian lässt sich darstellen: z.B. balkendiagramm auf Karte?

a) anzahl der registrierten asn pro land
b) anzahl der registrierten asn pro land, die auch aktiv sind
c) anzahl der asn pro land, die wir auf basis maxmind als tatsächlich im land aktiv ermittelt haben

info (a) ist eher wertlos - ist aber als Referenz auf die Zahl bei bgp.he.net zum Vergleich wichtig
Vergleich von (b) und (c) wäre interessant - weil (b) ist die übliche Zahl die man kennt (bei bgp.he.net im Länderreport die Anzahl der angezeigten ASN bzw. in radar.cloudflare.com detto) - und (c) ist die interessante Zahl nach der wir suchen … vielleicht ein Doppelbalkendiagram pro Land mit der Zahl (b) und (c) als Basis?

4)

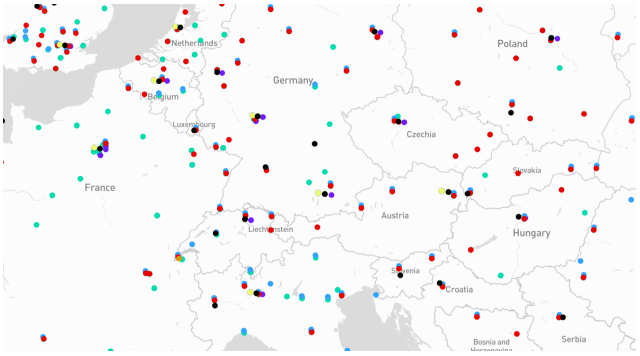
Wir können eine Summe der IPv4 und der IPv6 Prefixe pro Land summieren auf Basis der Zugehörigkeit zu 3b bzw. 3c -
D.h. wie viele der in einer Internet Full Table gefundenen Prefixanzahl ist laut registrierter ASN in einem Land bzw. wie viele der Prefixe bilden sich laut Meinung Maxmind in einem Land ab?

5)

Ich gehe davon aus, dass Maxmind nicht zu jedem gerouteten Eintrag eine Länderreferenz wissen wird? Wie viele unverortete Prefixe gibt es und zu welchem Land lassen sich diese aufgrund der allocation Referenz aufsummieren?
(d.h.: ungereimtheiten in den Daten geografisch mappen)

6)

https://opentelecomdata.org/cdns/



Können wir diese Daten selbst auch in unserem Bestand führen?
Daraus ergäben sich weitere interessante Daten die man graphen könnte:
Nämlich: in welchen Datacentern lassen sich diese CDN Nodes dann direkt verorten und welche IXPs haben diese dort in welcher Bandbreite angeschlossen - dann würde dieses Bild sehr viel mehr sagen …
und: man könnte diese Karte vermutlich auch noch anreichern mit der Information in welchem ASN diese Cache nodes eigentlich stehen - d.h. man könnte zwischen eigenen Nodes der Hyperscaler und jenen, wo es nur embedded Cache gibt unterscheiden.

Sorry, Text ist noch in Entwicklung - Nachtrag folgt - Danke Bernd

7) Email von Oliver an Sebastian vom 27.7.23

Hi Sebastian,

wenn wir uns morgen in der grossen Runde treffen, ist vermutlich die Zeit zu kurz fuer alle Details. Daher wollte ich vorab schonmal eine E-Mail schicken.

Fuer die naechsten Schritte:

- Du hast ja jetzt schon einen grossen Fundus an Related Work gesichert.

Die Related Work kann schonmal in das Related Work Kapitel der Arbeit.

- Der Fokus der Arbeit sollte die Auswertung der Registrierungsinfos und der Maxmind Geolocations zu den announcten Prefixen sein. Du solltest im naechsten Schritt schonmal langsam mit statistischen Auswertungen beginnen.

Dafuer waeren zwei Oberfragen relevant:

1. Welche AS'se sind in einem Land (Auswahl nach Registration *und* IP Geolocation). Da kann man dann ganz gute Statistiken und Plots mit Laenderverteilungen machen.

2. Wie gut stimmen die Registration Locations mit den IP Geolocations der Prefixe ueberein?

Wir wissen ja, was wir erwarten, aber das muss man statistisch zeigen.

Fuer Hypergiants/Tier-1/Anycast/... gibt es viele IP Geolocations zu einer Registration Location. Fuer lokale Netze (z.B. Uni Klagenfurt) duerften/sollten beide Locations uebereinstimmen.

Das interessante sind jetzt Statistiken ueber den gesamten Datensatz.

Baue dazu mal eine Metrik:

i. Maxmind auf jeden Prefix anwenden

ii. Ratio bilden: wieviel % der Prefixe haben die gleiche Registration Location wie die Prefix Geolocation?

Ich wuerde erwarten, dass diese Metrik bei einem Tier1/Anycast/Hypergiant/... gegen 0% (also sehr klein) tendiert und bei einem Netz wie der Uni Klagenfurt gegen 100%. Die Netze ueber diese Metrik zu Clustern waere sehr spannend. Dazu kann man auch noch Informationen aus der PeeringDB fuer die ~13k ASse in PeeringDB nutzen, die eine grobe Klassifikation fuer die Netze angibt (inbound / outbound heavy / ...). Kai kann das sicher aus den CAIDA PeeringDB DUmps extrahieren und dir eine Liste von Klassifkationen pro AS geben.

3. Aus Punkt 2 kommt noch ein halber Punkt danach: Wo funktioniert Maxmind garnicht? Und nutzt Maxmind dann wirklich die Location aus der Registrierungsinfo? Methodisch koennte man das beantworten, indem man Maxmind mit anderen Location DBs (z.B. RIPE oder IP2Loc) vergleicht und auf die Outlier schaut, wo Laender/Kontiente nicht uebereinstimmen.

Das waere der grobe Fahrplan. Wichtig ist, jetzt die Aufbereitung der Daten in JSON in der ersten Stufe langsam abzuschliessen und in die Auswertung zu gehen. Gib das JSON dann bitte auch an Kai. Ihr koennte das ja gegenseitig testen, wie gut die Formate funktionieren und zum weiteren Anreichern geeignet sind.

Die anderen Aspekte (DNS-name basiertes Router Geolocation, BGP Community Locations, ...) sind erstmal out of scope.

Oliver