# Akamai Position Paper for 2025
# IAB Workshop on IP Address Geolocation (ip-geo)

**Author:** Erik Nygren <nygren@akamai.com> / <erik+ietf@nygren.org>
with contributions from Rizwan Dhanidina
**Date:** Oct 1, 2025

## Overview and Use-Cases

IP:Geo location databases and systems are an important part of the Internet ecosystem. They have their inherent limitations, but they are not immediately replaceable. Akamai has our own IP:Geo location product ("EdgeScape") that we both sell directly, use internally, and have incorporated into a number of our other products and services. Akamai also publishes multiple public IP:Geo feeds in RFC 8805 format for address space that we control.

IP:Geo information is needed for cases where it is important to have an approximation of the coarse geographic location of a publically routable IP endpoint on the Internet, whether it be the IP of a client's network, a DNS resolver, a datacenter, a NAT exit point, or an IP being used to signal IP:Geo information. Example use-cases include:

- Akamai's CDN "Mapping" system uses IP:Geo as one of its many inputs for constructing a representation of Internet topology (noting that Internet topology bears some natural relationships to geographic topology at a coarse level, but is inherently different). This is one of the many factors we use to direct client DNS lookups to nearby servers based on either their nameserver IP address or the client IP prefix sent in EDNS0 Client Subnet (ECS). Our Mapping system also looks at the connecting IPs of clients and uses this as one of many inputs to our load balancing feedback control system.
- Servers implementing business logic based on coarse IP:Geo information, such as for our CDN service, allows customers to implement custom business logic based on the coarse geo derived from an IP:Geo lookup of the connecting Client IP. Akamai has features where customers can match on client geo during HTTP request processing, send client geo information forward to origin, match on client geo during DNS lookups, etc. Customers use this abstraction for a wide range of purposes, within the limitations of IP:Geo systems. Some customer uses include redirecting to regional web sites, implementing legally or contractually mandated block lists, or providing results customized by geography.
- Enabling internal and customer-facing reporting, diagnostics, telemetry, and intelligence that is keyed on the geographic distribution of clients, such as the mix of traffic across country and subdivision. This aggregate information is used for a wide mixture of operational and business planning purposes.

Inherent in many of these is the need to be able to instantly map from IP address to geo information in an IP:Geo data structure, or to use the topology inherent in an IP:Geo topology tree as an input to processing other trees of IP information. While some of these uses (specifically in the case of customer/service business logic) might be amenable to other ways of obtaining client geographic information, the majority of these cases need IP:Geo and have made trade-offs against its limitations and gaps. Solutions that need more precise information (such as when a user opts-in to get highly localized search results, or for selecting a language/locale) are better off using alternative approaches, only falling back to IP:Geo when information from those other approaches are unavailable.

Akamai's EdgeScape constructs its database from a wide variety of sources, including other self-published RFC 8805 feeds. From the collected data, it evaluates data quality, fixes issues, and synthesizes an IP:Geo tree structure.

From our perspective there is significant value in having self-published IP:Geo feeds as this creates an ecosystem where network operators and IP:Geo services have a way to exchange information. As the operator of both a CDN and a cloud platform we have found substantial value in publishing IP:Geo feeds for our own IP space as this allows other IP:Geo systems to properly locate our datacenters for their use-cases.

However, our experience is that self-published feeds are not adequate on their own. Aggregators such as the EdgeScape product, as well as other commercial services and free data products, are needed to synthesize from various sources, validate and clean data, and to provide a database that can be used for highly efficient and rapid lookups.

# Gaps and Problems with IP:Geo

Some problems with the current IP:Geo ecosystem include:

- Many networks do not provide self-published feeds, leaving IP:Geo database systems to use other sources of information as inputs. This can cause problems for their users, especially in cases where traffic shows up from newly allocated or recently relocated IP space.
- Self-published feeds often have errors or availability issues. We download hundreds per week based on the geofeed entries in RDAP and see that feeds go offline or go missing, and some feeds sometimes have incorrect entries (cities in one state when they should be in another, cities spelled incorrectly, badly formatted entries, etc).
- There can be significant latency between publishing an update to a self-published feed and having it get picked up and redistributed by various IP:Geo services. This can be challenging for network operators who need to restructure their IPv4 space or deploy into new locations.
- Self-published feeds in RFC 8805 format do not have a way to specify the level of precision for entries. For situations like Anycast IPs, for aggregates, or other cases that are inherently very coarse, having a way to provide information about precision or radius could add value.

There are also issues inherent to IP:Geo, including:

- In certain network deployments, IP:Geo can be extremely coarse (eg, at best country-level). Some broadband operators assign endpoints as IPv4 /32 and remap with an internal overlay at their border, meaning that their address structure does not convey geographic topology.
- Other networks use CGNAT or IPv4aaS technologies such as MAP-T meaning that IPv4 traffic egresses centrally through a NAT. In these cases IPv6 can have more accuracy and precision than IPv4 (which on the plus side should encourage more content providers to enable IPv6!).
- IP:Geo can have different results for a given client for IPv4 vs IPv6. This can be a problem when an authentication token is bound on issuance by IPv4 geo and validated by IPv6 geo, or vice-versa.
- It is impossible for IP:Geo to be 100% accurate. Malicious users can always use VPNs, compromised hosts, open proxies, and other services to bypass IP:Geo based policy controls. Incorrect IP:Geo may also result in some portion of users getting blocked or incorrect content. While IP:Geo is sometimes mandated as a layer of protection, it is not necessarily sufficient on its own.

IP:Geo databases are also inherently bounded in size: with IPv4 there would never be more than 4 billion entries (eg, 4GB with a byte per IP). Even the size of an IPv6 database is practically limited by the scalability of routing tables. While this has not been a problem for cases where IP:Geo is appropriate (and conveniently bounds the problem when taking an IP:Geo approach), it can be a further motivation to use other approaches where precise location or user-influenced preference are more appropriate.

# Relationship to other Problem Spaces

While other systems such as IP Reputation may seem similar in nature to IP:Geo, they are fundamentally different and with different incentives. While IP:Geo has a “dense tree” representation mapping to geo, IP Reputation tends to be much more sparse and with highly dimensional attributes. IP Reputation systems also face a very different adversarial model and are much more dynamic. In this space draft-ietf-opsawg-prefix-lengths can add value by allowing networks to express the structure of IP space.

Adding attributes to IP:Geo feeds — such as whether IP space is a cloud hosting provider, mobile network, broadband network, etc — might have some value. Some attributes are not inherent in the IP structure however (eg, wired and wireless clients are often intermixed in the same IP space).

Having a way to exchange IP ACL feeds between parties would be extremely useful to standardize, but it is also a distinctly different problem space.