CDS Spec: Lacuna Feedback

Audience: OMF CDS Working Group
Status: first draft, in progress (July 2021)

Purpose of this document

Suggest 3 changes to the existing CDS Spec. that, if adopted, would

- Allow providers to fetch curb zone regulations without their associated geometries, improving scalability and easing provider integration
- Make CDS policies/regulations auditable by a city user
- Keep CDS consistent and interoperable with MDS

Proposed Changes

The three-part changes involve:

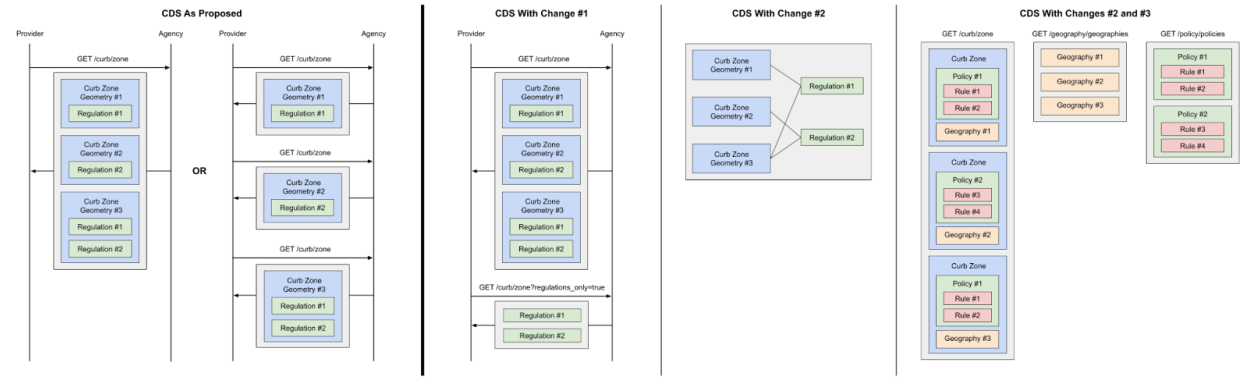
1. Separating regulations and geometry, both in logic and in how they can be fetched
2. Promoting the (now-separate) regulations as separate objects
3. Defining the (now-separate) geometry and regulation information as MDS JSON objects

Change		Description	Motivation
#1	Separate Regulations and Geometry	Make regulations available without GeoJSON using a query parameter flag on GET /curb/zone endpoints	Allow providers to fetch regulations without their associated geometries, improving scalability
#2	Make regulations separate objects, exposed in own endpoint	Store Curb Zones separately from regulations	• Establish a many-to-many relationship between Curb Zones and regulations reduces data duplication in cases where the same regulation applies to many Curb Zones • More easily enable regulation modifications, publishing and unpublishing regulations, and querying historical regulations • Enable auditability of CDS policies/regulations by a city user
#3 (Must be implemented with #2)	Define Curb Zones as JSON MDS objects	• Curb Zones are JSON objects with (MDS Geography) GeoJSON properties • Rules about the curb are encoded as MDS Policy instead of Regulations	• Keep CDS consistent and interoperable with MDS, easing provider integration

Impacts of Implementing Proposed Changes

Example		Change #1	Change #2	Change #2+3
A	GIS Tool Use	• No impact	• No impact when a full curb zone GeoJSON is fetched	• Curb Zones no longer directly viewable in GIS tools
B	Integrating Providers	• Better scalability with curb zone complexity and quantity, as well as frequency of regulation changes • Providers could fetch curb zone geometries infrequently and cache them and fetch regulations more frequently, reducing latency	• Enable independent scaling of the infrastructure powering each independent API	• Ease provider integration by eliminating need to understand new concept of Regulations • Any existing tools built by providers around MDS Geography and MDS Policy could be reused
C	Auditing by a City User	• No impact	• Richer available metadata on regulations, enabling auditability and analytics use cases	• Ease agency adoption by eliminating need to understand new concept of Regulations • Any existing tools built by agencies around MDS Geography and MDS Policy could be reused

Summary of Proposed Changes



Detailed Comparison of Proposed Changes

CDS Regulations and Zones Vs. MDS Policy/Rules and Geography

CDS must encode several types of information to accomplish its goal of enabling the management of short term loading activity in urban curb space. These types of information include:

- The physical characteristics of the curb space.
- The rules governing use of the space.
- The target audience the rules apply to.
- The time the rules are in effect.

CDS proposes encoding these four types of information through the introduction of four new types of data structures: `CurbZones`, `CurbRegulations`, `user_classes`, and `TimeSpans`. These are new objects introduced specifically for CDS, and differ from the objects MDS generally uses to model these things (`Geographies`, `Policies`, `vehicle_types` / `propulsion_types` / `provider_ids`, and various `Policy` properties, respectively).

Physical Curb Space

In CDS, the physical characteristics of the curb space include its GeoJSON and linear reference representations, its length, cross streets, etc. In MDS, geographic location information like this is represented generally through the Geography spec.

This information is relatively static and may change at the rate of roughly once every 6+ months.

Representation of physical curb space	

CDS: Curb Zone object (a GeoJSON Polygon feature with specific expected properties)	<pre>{
	  type: "Feature",
	  id: UUID,
	  geometry: {
	  type: "Polygon",
	  coordinates: Number[][][]
	  },
	  properties: {
	  regulations: CurbRegulation[],
	  effective_date: String,
	  end_date?: String,
	  location_references?: LocationReference[],
	  name?: String,
	  zone_id?: UUID,
	  street_name?: String,
	  cross_street_start_name?: String,
	  cross_street_end_name?: String,
	  length?: Number,
	  available_space_lengths?: Number[],
	  availability_time?: String,
	  width?: Number,
	  parking_angle?: Enum,
	  num_spaces?: Integer,
	  street_side?: Enum,
	  median?: Boolean,
	  entire_roadway?: Boolean,
	  curb_area?: UUID
	  }
	}</pre>

MDS: Geography object	<pre>{
	  name: String,
	  description?: String,
	  geography_type?: String,
	  geography_id: UUID,
	  geography_json: GeoJSONFeatureCollection,
	  effective_date?: Timestamp,
	  published_date: Timestamp,
	  retire_date?: Timestamp,
	  prev_geographies?: UUID[]
	}</pre>

Rules

The curb regulation object must specify the type of activity to be regulated and the value of the limit itself (e.g., “no stopping for more than 15 minutes”).

A CDS Curb Regulation object is a rule that allows or prohibits a particular set of users from using a particular curb at a particular time or times. Regulations are inspired by curbLR. A regulation consists of a single rule to be enforced, an integer `priority`, a list of `user_classes` the rule applies to, and the `time_spans` during which the regulation is in effect. In MDS, agency regulations are encoded as `Policy` objects which are composed of `Rules`.

This information is relatively dynamic.

Representation of rules	

CDS: CurbRegulation object	<pre>{
	  policy_id: UUID,
	  priority: Integer,
	  rule: {
	  activity: String,
	  max_stay?: Integer,
	  no_return?: Integer
	  },
	  user_classes?: Enum[],
	  time_spans?: TimeSpan[],
	  rate?: {
	  per_hour: Integer,
	  start_minutes?: Integer,
	  end_minutes?: Integer
	  }[]
	}</pre>

MDS: Policy object	<pre>{
	  name: String,
	  policy_id: UUID,
	  provider_ids?: UUID[],
	  description: String,
	  currency?: String,
	  start_date: Timestamp,
	  end_date?: Timestamp,
	  published_date: Timestamp,
	  prev_policies?: UUID[],
	  rules: {
	  name: String,
	  rule_id: UUID,
	  rule_type: Enum,
	  geographies: UUID[],
	  states: Enum,
	  rule_units?: Enum,
	  vehicle_types?: Enum[],
	  propulsion_types?: Enum[],
	  minimum?: Integer,
	  maximum?: Integer,
	  rate_amount?: Integer,
	  rate_recurrence?: Enum,
	  start_time?: String,
	  end_time?: String,
	  days?: Enum[],
	  messages?: String[],
	  value_url?: String
	  }[]
	}</pre>

Rule Audience

The rule audience could be a specific provider or a vehicle type (e.g., all motorcycles, or trucks over 5 tons). In CDS, regulations can target one or more `user_classes`. In MDS, policies (and individual rules within policies) can target any combination of `provider_id`, `vehicle_type`, and/or `propulsion_type`.

Representation of rule audience	
CDS: user_classes	`enum User_Classes { bicycle, bus, commercial, handicap, motorcycle, official, permit, rideshare, taxi, truck }`

MDS: provider_ids, vehicle_types, propulsion_types	`enum Vehicle_Types { car, bicycle, scooter, moped, other }`
	
	`enum Propulsion_Types { human, electric, electric_assist, hybrid, combustion }`

Rule Time Period

Regulations and Policies alike may have an absolute start and end time (e.g., "Jan 1, 2020 through Dec 31, 2022") as well as a recurring time period it is in effect (e.g., “weekdays after 5pm” or “2nd Tuesday of every month”). In CDS, this information is stored as one or more `TimeSpan` properties on the `CurbRegulation` object. In MDS, this information is stored in various properties of the `Policy` and `Rule` objects.

Representation of rule time period	

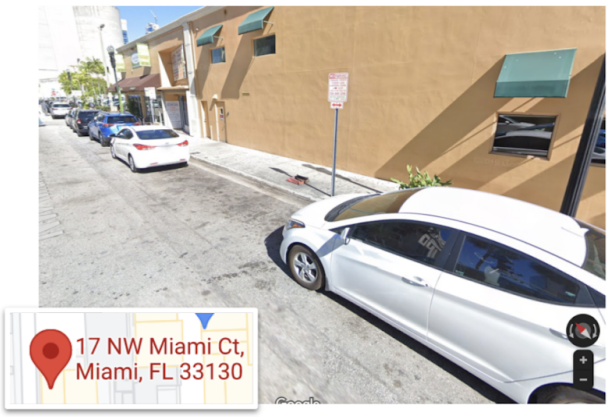
CDS: TimeSpan property on CurbRegulation object	`{`
	`  from?: String,`
	`  to?: String,`
	`  days_of_week?: Enum[],`
	`  months?: Integer[],`
	`  time_of_day_start?: String,`
	`  time_of_day_end?: String,`
	`  designated_period?: String,`
	`  designated_period_except?: Boolean`
	`}`

MDS: Policy and Rule object properties (policies are composed of rules)	Policy:
	`start_date: Timestamp`
	`end_date?: Timestamp`
	
	Rule:
	`start_time?: String`
	`end_time?: String`
	`days?: Enum[]`

Comparing CDS vs. CDS + Changes

We will now walk through the process of encoding the location and regulations for a real curb zone, exploring how its location and rules can be modeled with CDS and what that would look like with the three proposed changes.

	A	B	D	E	F	G
1	Buliding Name & Address	Loading Zone Type	Side of Street	Time & Day	Length	Metered Zone
7	Starbucks Coffee 17 NW Miami Court	Freight No Parking Trucks 7000lbs 3.5T (Flex)	East Side	7am-6pm Mon-Fri	60ft	703



The above scenario can be found in downtown Miami. The regulation specifies that this 60' curb zone be used for commercial loading and unloading between 7am and 6pm on weekdays and for passenger paid parking outside of those hours on weekdays. Additionally, it is always restricted to trucks over 7,000 lbs.

Rules

- No parking for trucks 7000lbs 3.5T | Every day
- Reserved for commercial loading and unloading | 7:00am-6:00pm | Every week day
- Reserved for passenger parking | 6:00pm-7:00am | Every week day | 3 hour limit, $2 /hr

We will walk through the impact of the proposed changes for three scenarios

1 - Interoperability with GIS tools
2 - Provider integration
3 - Auditability by a City User

CDS Representation

CDS, as currently proposed, would represent this zone roughly as follows:

```
{
  type: "Feature",
  id: "045f5d7e-3064-42a1-9ede-f1f22908f2ba",
  geometry: {
    type: "Polygon",
    coordinates: [
      [
        [-80.1935890, 25.7760446],
        [-80.1935541, 25.7751460],
        [-80.1935943, 25.7751460],
        [-80.1936292, 25.7760470],
        [-80.1935890, 25.7760446]
      ]
    ]
  },
  properties: {
    regulations: [
      // No trucks over 7,000 lbs
      {
        policy_id: "642a92d2-7f44-42f2-859a-75e51d7f9943",
        priority: 0,
        rule: { activity: "no parking" },
        user_classes: [ "trucks_7000lbs" ]
      },
      // Commercial loading 7am-6pm weekdays
      {
        policy_id: "5a0057d1-1bc4-4122-b788-c3af161b958f",
        priority: 1,
        rule: { activity: "loading" },
        user_classes: [ "commercial" ],
        time_spans: [
          {
            time_of_day_start: "07:00",
            time_of_day_end: "18:00",
            days_of_week: [ "mon", "tue" ,"wed", "thu", "fri" ]
          }
        ]
      },
      // Paid passenger parking 6pm-7am weekdays
      {
        policy_id: "a845a523-eec0-4a05-bb6e-b05434389475",
        priority: 3,
        rule: { activity: "parking", max_stay: 180 },
        user_classes: [ "car" ],
        time_spans: [
          {
            time_of_day_start: "18:00",
            time_of_day_end: "07:00",
            days_of_week: ["mon", "tue" ,"wed", "thu", "fri"]
          }
        ],
        rate: [
```

```
          { per_hour: 200 }
        ]
      }
    ],
    effective_date: "2019-10-12T07:20:50.52Z",
    location_references: [
      {
           source: "https://sharedstreets.io",
           ref_id: "jwwKcUvHuCw6GJJAT3mDQ",
           start: 9.1,
           end: 22.4,
           side: "right"
      }
    ],
    name: "Starbucks Coffee 17 NW Miami Court",
    street_name:"NW Miami Ct",
    cross_street_start_name: "W Flagler St",
    cross_street_end_name: "NW 1st St",
    length: 33.3,
    num_spaces: 6,
    street_side: "E",
    median: false,
    entire_roadway: false,
    curb_area: "6283faf1-d8f6-4205-8a6e-7a883f24e714"
  }
}
```

Impact of Change #1 ONLY:
Make Regulations Available Without GeoJSON

Curb geometries and curb regulations have different lifecycles; a curb's geometry is largely fixed and may only change once or twice a year. A curb's regulations, on the other hand, could conceivably change several times a month as curb space is reallocated for things like special events and roadwork. Moreover, digital expression of curb regulations will enable significantly more dynamic regulations in the future - for example, the real-time adjustment of passenger parking rates based on demand.

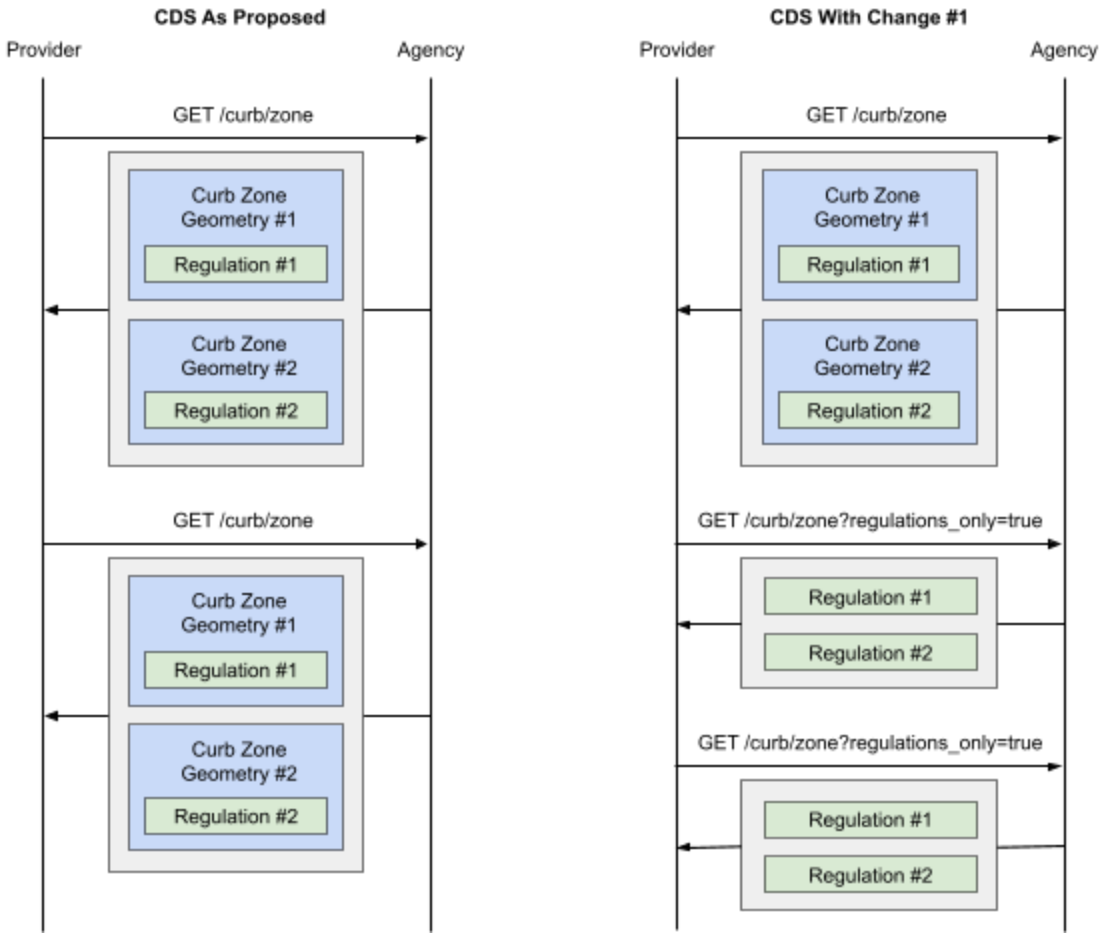
Given the relative difference in lifecycles between curb zones and regulations, the CDS API will scale better with curb zone complexity and quantity as well as with frequency of regulation changes **if regulations can be accessed separately from curb zones**. Increasing curb zone complexity and quantity will increase the size of the curb zone geometry part of the GET /curb/zone endpoint response, and frequently changing regulations (like demand-responsive pricing) will require providers to poll for regulation changes equally frequently. These two dynamics work against each other somewhat in the current design where regulations and curb zone GeoJSON are always bundled. However, this risk could be alleviated with the addition of a single query parameter flag on the proposed GET /curb/zone endpoints that would allow providers to fetch curb zone regulations without their associated geometries.

Use Case 1: Interoperability with GIS tools

This proposed change would not affect the ability for users to fetch full curb zone GeoJSON objects as needed and load them directly into a GIS viewer tool, to be rendered natively as GeoJSON.

Use Case 2: Provider Integration

As discussed above, providers would benefit from the ability to fetch regulations frequently to reduce the latency of those regulation changes reaching their routing and dispatch systems. With this change, they would be able to fetch curb zone geometries infrequently, cache them, and then frequently fetch curb zone regulations without their associated geometries which are already known.




```
GET /curb/zone/045f5d7e-3064-42a1-9ede-f1f22908f2ba?regulations_only=true

[
  // No trucks over 7,000 lbs
  {
    policy_id: "642a92d2-7f44-42f2-859a-75e51d7f9943",
```

```
    priority: 0,
    rule: { activity: "no parking" },
    user_classes: [ "trucks_7000lbs" ]
  },
  // Commercial loading 7am-6pm weekdays
  {
    policy_id: "5a0057d1-1bc4-4122-b788-c3af161b958f",
    priority: 1,
    rule: { activity: "loading" },
    user_classes: [ "commercial" ],
    time_spans: [
      {
        time_of_day_start: "07:00",
        time_of_day_end: "18:00",
        days_of_week: [ "mon", "tue" ,"wed", "thu", "fri" ]
      }
    ]
  },
  // Paid passenger parking 6pm-7am weekdays
  {
    policy_id: "a845a523-eec0-4a05-bb6e-b05434389475",
    priority: 3,
    rule: { activity: "parking", max_stay: 180 },
    user_classes: [ "car" ],
    time_spans: [
      {
        time_of_day_start: "18:00",
        time_of_day_end: "07:00",
        days_of_week: ["mon", "tue" ,"wed", "thu", "fri"]
      }
    ],
    rate: [
      { per_hour: 200 }
    ]
  }
]
```

Use Case 3: Auditability by a City User

A city employee who wishes to audit historic curb zones and regulations will be unaffected by this change.

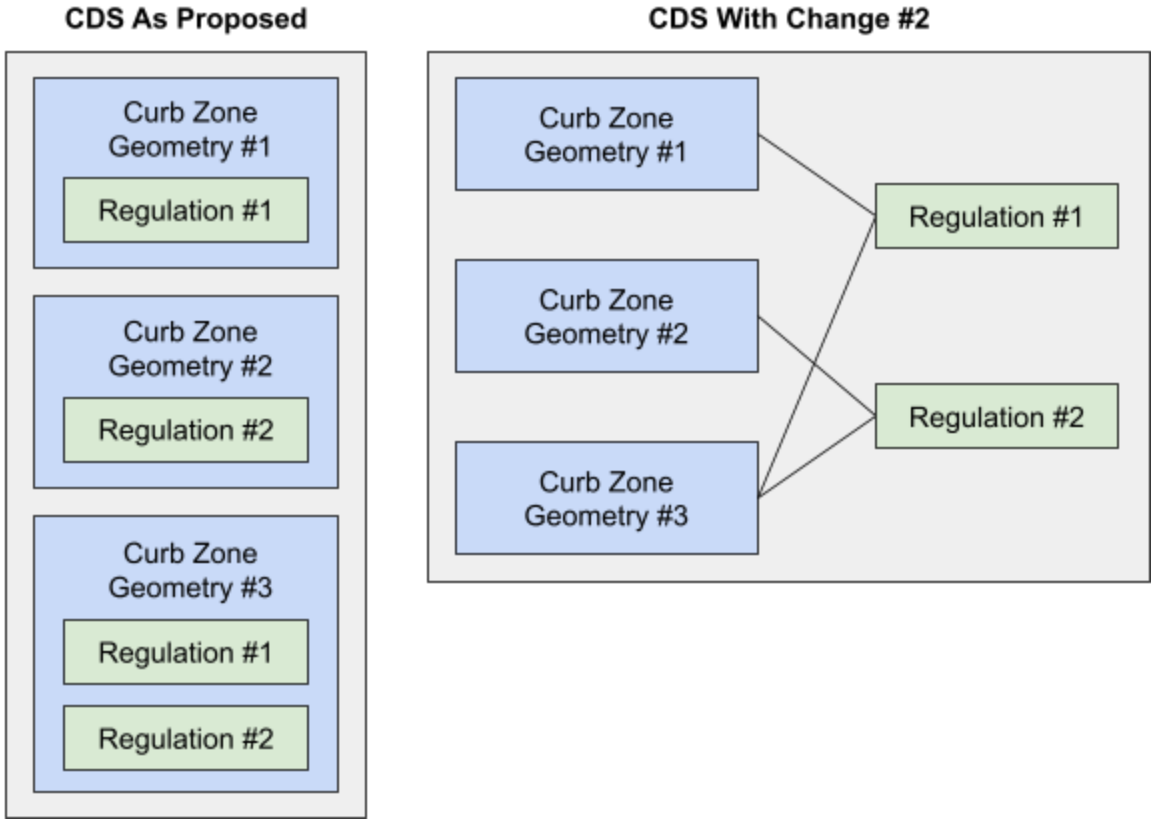
Impact of Change #2 ONLY: Treat Regulations as Separate Objects

Regulations are critical entities in the curb ecosystem and it is easy to imagine city user use cases that would involve regulations directly as first-class entities. For example, a city employee may need to look at records of when regulations were created and by whom and when they were published in order to resolve a disputed curb usage fees or fines. Similarly, a city employee may be interested in the effects of their curb regulation policies on nearby parking tickets and would need access to information on when various regulations went into effect.

These use cases could be satisfied with the addition of metadata like author, created_date, start_date, and end_date to regulations.

Exposing regulations in their own endpoint - going one step further than the query flag approach described above - would also provide space for more advanced regulation operations in the future, like modifying regulations, publishing and unpublishing regulations, or querying for historical regulations.

Additionally, if regulations are stored separately from CurbZones, a many-to-many relationship between regulations and CurbZones can be established, reducing data duplication in cases where the same regulation applies to many CurbZones (e.g., all "Passenger loading only" CurbZones in a given neighborhood).



Separating Regulations from CurbZones in this way would allow the API response to reference Regulations by their regulation_id instead of duplicating the full regulation within each CurbZone object.

Use Case 1: Interoperability with GIS tools

This proposed change would not affect the ability for users to fetch full curb zone GeoJSON objects as needed and load them directly into a GIS viewer tool (along with their regulations), to be rendered natively as GeoJSON.

Use Case 2: Provider Integration

A dedicated endpoint for fetching regulations separately from the physical curb space would enable the backend APIs supporting these two different data types to scale independently, assuming they would be subject to different request loads from providers. Such an endpoint could look something like:

```
GET /curb/regulations/642a92d2-7f44-42f2-859a-75e51d7f9943

{
  author: "agency.employee@dot.gov",
  created_date: 1612699200000,
  start_date: 1613304000000,
  end_date: 1644840000000,
  rules: [
    // No trucks over 7,000 lbs
    {
      policy_id: "642a92d2-7f44-42f2-859a-75e51d7f9943",
      priority: 0,
      rule: { activity: "no parking" },
      user_classes: [ "trucks_7000lbs" ]
    },
    // Commercial loading 7am-6pm weekdays
    {
      policy_id: "5a0057d1-1bc4-4122-b788-c3af161b958f",
      priority: 1,
      rule: { activity: "loading" },
      user_classes: [ "commercial" ],
      time_spans: [
        {
          time_of_day_start: "07:00",
          time_of_day_end: "18:00",
          days_of_week: [ "mon", "tue" ,"wed", "thu", "fri" ]
        }
      ]
    },
    // Paid passenger parking 6pm-7am weekdays
    {
      policy_id: "a845a523-eec0-4a05-bb6e-b05434389475",
      priority: 3,
      rule: { activity: "parking", max_stay: 180 },
      user_classes: [ "car" ],
      time_spans: [
        {
          time_of_day_start: "18:00",
          time_of_day_end: "07:00",
          days_of_week: ["mon", "tue" ,"wed", "thu", "fri"]
        }
      ],
      rate: [
        { per_hour: 200 }
      ]
    }
  ]
}
```

Use Case 3: Auditability by a City User

Knowing metadata such as the `author`, `created_date`, `start_date`, and `end_date` of each regulation would enable auditability and analytics use cases on regulations as discussed above. Without this information, answering questions from commercial operators like "Why was I charged a $5 parking fee?" or from agency decision makers like "What impact are the new regulations having on our organization KPIs?" may be challenging in some cases.

Impact of Change #2+3 COMBINED: Separate Geometry + Regulation; Define Curb Zones as JSON MDS Objects; Define Regulations as Policy

What if in addition to Change #2, Curb Zones were JSON objects with an MDS Geography property and rules about the curb were encoded as MDS Policy instead of Regulations?

With the addition of metadata and a separate endpoint to regulations as discussed in Change #2, regulations would look even more similar to Policies than they do in the proposed CDS spec. Any providers who are already familiar with Policies through their exposure to MDS may be confused by the introduction of the similar concept of Regulations. Additionally, any infrastructure providers have already built to consume or otherwise interact with Policies will not be reusable for Regulations. Ease of provider integration with CDS would be improved if the curb rules are encoded as Policies instead of Regulations. This would also support the OMF's charter goal to ensure interoperability with other technology managed by the OMF, to avoid duplication and downstream implementation complexity.

Additionally, defining CurbZones as GeoJSON objects at the top level may also introduce confusion to providers. GeoJSON is a well known and commonly used general standard for defining arbitrary geospatial data. Defining a CurbZone as GeoJSON doesn't provide much immediate context to someone trying to learn what a CurbZone is, as GeoJSON (by its general nature) is used to describe everything from administrative boundaries to point locations and road curvatures. Of course CurbZones are not only GeoJSON, they are GeoJSON but with specific expected metadata properties - a concept that could be slightly cumbersome to explain and express digitally.

Alternatively, CurbZones could be defined as their own kind of JSON object - immediately indicating to the user by their type definition that they are more than some special kind of GeoJSON - they're CurbZone objects. Of course they could still have a GeoJSON property to encode the curb zone's geography, but they could also have top-level metadata properties like `author` which then wouldn't have to exist below the CurbZone's geometry hierarchically.

Use Case 1: Interoperability with GIS tools

This change would break the ability for CurbZone objects to be loaded into various GIS viewer tools directly for viewing on a map, as they would no longer be GeoJSON but rather custom JSON objects. However, a simple script could extract the geography properties of each CurbZone for viewing in a GIS tool. Alternatively, a query parameter on the GET /curb/zone endpoints could potentially format the response as GeoJSON (or JSON).

Use Case 2: Provider Integration

If curb zones used MDS Geography to represent their geographic locations and MDS Policy to represent their rules and regulations, a provider's integration to fetch curb zones and policies could look something like this:

```
GET /curb/zone/045f5d7e-3064-42a1-9ede-f1f22908f2ba

{
  curb_zone_id: "045f5d7e-3064-42a1-9ede-f1f22908f2ba",
  curb_zone_name: "Starbucks Coffee 17 NW Miami Court",
  street_name: "NW Miami Ct",
  cross_street_start_name: "W Flagler St",
  cross_street_end_name: "NW 1st St",
  length: 33.3,
  num_spaces: 6,
  street_side: "E",
  median: false,
  entire_roadway: false,
  curb_area: "6283faf1-d8f6-4205-8a6e-7a883f24e714",
  effective_date: "2019-10-12T07:20:50.52Z",
  location_references: [
    {
      source: "https://sharedstreets.io",
      ref_id: "jwwKcUvHuCw6GJJAT3mDQ",
      start: 9.1,
      end: 22.4,
      side: "right"
    }
  ],
  geography: {
    geography_id: "b507705a-14cb-4fb7-a9d6-d5b0db1b00d4",
    name: "Starbucks Coffee 17 NW Miami Court",
    effective_date: 162395418000,
    publish_date: 1623954171281,
    prev_geographies: null,
    geography_json: {
      type: "FeatureCollection",
      features: [
        {
          type: "Feature",
          properties: {},
          geometry: {
            type: "Polygon",
            coordinates: [
              [
```

```
              [-80.1935890, 25.7760446],
              [-80.1935541, 25.7751460],
              [-80.1935943, 25.7751460],
              [-80.1936292, 25.7760470],
              [-80.1935890, 25.7760446]
            ]
          ]
        }
      }
    ]
  }
},
policy: {
  name: "No Heavy Trucks, Paid Passenger Parking",
  description: "Curb Policy At Starbucks 17 NW Miami Court",
  policy_id: "4ca4f6c1-db92-4884-b56a-ee6dffed86d7",
  start_date: 1603834230000,
  end_date: 1603852200000,
  geographies: [ "5aac436a-33eb-4754-b41d-f8a3800569c0" ],
  rules: [
    {
      name: "No heavy vehicle zone",
      rule_id: "a19671dc-e5ae-4133-b56a-5d6ef984bf3d",
      rule_type: "prohibition",
      start_time: "7:00:00",
      end_time: "18:00:00",
      maximum: 0,
      vehicle_gvwr_lim: [
        "3.5 tons"
      ]
    },
    {
      name: "Commercial loading 7am-6pm weekdays",
      rule_id: "652d0650-f9a3-45d3-9a81-18bba831fb29",
      rule_type: "prohibition",
      start_time: "7:00:00",
      end_time: "18:00:00",
      maximum: 0,
      days: ["mon", "tue", "wed", "thu", "fri"],
      geographies: [
        "5aac436a-33eb-4754-b41d-f8a3800569c0"
      ],
      vehicle_types: ["car", "bicycle", "scooter", "moped"]
    },
    {
      "name": "Passenger paid parking only",
      "rule_id": "1c281903-835b-46b4-af47-8ef41241dda4",
      "rule_type": "dwell_time",
      "start_time": "18:00:00",
      "end_time": "7:00:00",
      "days": ["mon", "tue", "wed", "thu", "fri"],
      "maximum": 3,
      "rule_unit": "hour"
      "vehicle_types": ["car"]
    },
    {
      "name": "Passenger parking fee",
```

```
        "rule_id": "70f7593f-df40-4395-b3fa-eecc45179ce5",
        "rule_type": "dwell_time",
        "start_time": "18:00:00",
        "end_time": "7:00:00",
        "days": ["mon", "tue", "wed", "thu", "fri"],
        "rate_amount": 2,
        "currency": "USD",
        "rule_unit": "hour",
        "vehicle_types": ["car"],
        "statuses": {
          "parked"
        }
      }
    ]
  }
}
```

```
GET /geography/geographies

[
  {
    geography_id: "b507705a-14cb-4fb7-a9d6-d5b0db1b00d4",
    name: "Starbucks Coffee 17 NW Miami Court",
    effective_date: 162395418000,
    publish_date: 1623954171281,
    prev_geographies: null,
    geography_json: {
      type: "FeatureCollection",
      features: [
        {
          type: "Feature",
          properties: {},
          geometry: {
            type: "Polygon",
            coordinates: [
              [
                [-80.1935890, 25.7760446],
                [-80.1935541, 25.7751460],
                [-80.1935943, 25.7751460],
                [-80.1936292, 25.7760470],
                [-80.1935890, 25.7760446]
              ]
            ]
          }
        }
      ]
    }
  }
]
```

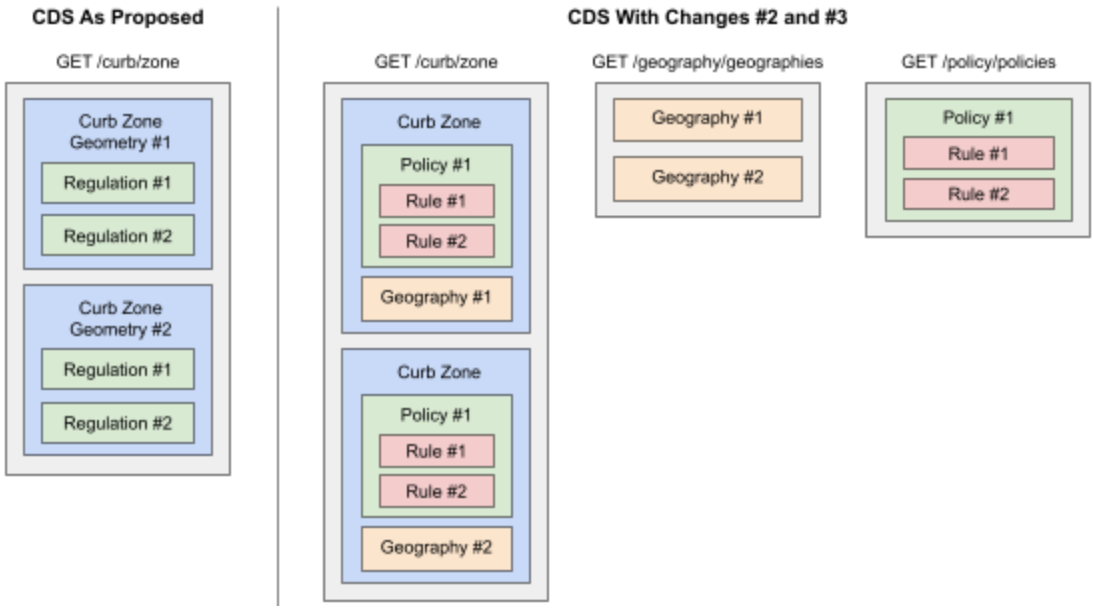
```
GET /policy/policies

[
  {
    name: "No Heavy Trucks, Paid Passenger Parking",
```

```
  description: "Curb Policy At Starbucks 17 NW Miami Court",
  policy_id: "4ca4f6c1-db92-4884-b56a-ee6dffed86d7",
  start_date: 1603834230000,
  end_date: 1603852200000,
  geographies: [ "5aac436a-33eb-4754-b41d-f8a3800569c0" ],
  rules: [
    {
      name: "No heavy vehicle zone",
      rule_id: "a19671dc-e5ae-4133-b56a-5d6ef984bf3d",
      rule_type: "prohibition",
      start_time: "7:00:00",
      end_time: "18:00:00",
      maximum: 0,
      vehicle_gvwr_lim: [
        "3.5 tons"
      ]
    },
    {
      name: "Commercial loading 7am-6pm weekdays",
      rule_id: "652d0650-f9a3-45d3-9a81-18bba831fb29",
      rule_type: "prohibition",
      start_time: "7:00:00",
      end_time: "18:00:00",
      maximum: 0,
      days: ["mon", "tue", "wed", "thu", "fri"],
      geographies: [
        "5aac436a-33eb-4754-b41d-f8a3800569c0"
      ],
      vehicle_types: ["car", "bicycle", "scooter", "moped"]
    },
    {
      "name": "Passenger paid parking only",
      "rule_id": "1c281903-835b-46b4-af47-8ef41241dda4",
      "rule_type": "dwell_time",
      "start_time": "18:00:00",
      "end_time": "7:00:00",
      "days": ["mon", "tue", "wed", "thu", "fri"],
      "maximum": 3,
      "rule_unit": "hour"
      "vehicle_types": ["car"]
    },
    {
      "name": "Passenger parking fee",
      "rule_id": "70f7593f-df40-4395-b3fa-eecc45179ce5",
      "rule_type": "dwell_time",
      "start_time": "18:00:00",
      "end_time": "7:00:00",
      "days": ["mon", "tue", "wed", "thu", "fri"],
      "rate_amount": 2,
      "currency": "USD",
      "rule_unit": "hour",
      "vehicle_types": ["car"],
      "statuses": {
        "parked"
      }
    }
  ]
```

```
    }
]
```

Note that the Curb Zone's MDS Geography and MDS Policy objects could still be included in full for each Curb Zone in the GET /curb/zones endpoint (even though they are managed by the MDS Geography and MDS Policy backend services respectively). **This would ease provider integration by eliminating the need to understand the new concept of Regulations, and allowing the reuse of any existing tools built by providers around MDS Geography and MDS Policy.**



Use Case 3: Auditability by a City User

Aside from the GIS tool integration for viewing CurbZones natively, the auditability and analytics use cases discussed above would not be negatively affected by this change. In fact, any existing tools, workflows, or infrastructure built up by agencies around MDS Geography and MDS Policy could be reused to apply to the curb.