

iab.TECH LAB

Buyers.json and DemandChain Object Implementation Guide

Released June 2021

Please email support@iabtechlab.com with feedback or questions. This document is available online at <https://iabtechlab.com/buyers-json-demand-chain>

© IAB Technology Laboratory

About IAB Tech Lab

The IAB Technology Laboratory is a nonprofit research and development consortium charged with producing and helping companies implement global industry technical standards and solutions. The goal of the Tech Lab is to reduce friction associated with the digital advertising and marketing supply chain while contributing to the safe growth of an industry.

The IAB Tech Lab spearheads the development of technical standards, creates and maintains a code library to assist in rapid, cost-effective implementation of IAB standards, and establishes a test platform for companies to evaluate the compatibility of their technology solutions with IAB standards, which for 18 years have been the foundation for interoperability and profitable growth in the digital advertising supply chain. Further details about the IAB Technology Lab can be found at <https://iabtechlab.com>.

License

Buyers.json by the IAB Tech Lab's Programmatic Standards Working Group is licensed under a [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/). To view a copy of this license, visit creativecommons.org/licenses/by/3.0/ or write to Creative Commons, 171 Second Street, Suite 300, San Francisco, CA 94105, USA.



Disclaimer

THE STANDARDS, THE SPECIFICATIONS, THE MEASUREMENT GUIDELINES, AND ANY OTHER MATERIALS OR SERVICES PROVIDED TO OR USED BY YOU HEREUNDER (THE "PRODUCTS AND SERVICES") ARE PROVIDED "AS IS" AND "AS AVAILABLE," AND IAB TECHNOLOGY LABORATORY, INC. ("TECH LAB") MAKES NO WARRANTY WITH RESPECT TO THE SAME AND HEREBY DISCLAIMS ANY AND ALL EXPRESS, IMPLIED, OR STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AVAILABILITY, ERROR-FREE OR UNINTERRUPTED OPERATION, AND ANY WARRANTIES ARISING FROM A COURSE OF DEALING, COURSE OF PERFORMANCE, OR USAGE OF TRADE. TO THE EXTENT THAT TECH LAB MAY NOT AS A MATTER OF APPLICABLE LAW DISCLAIM ANY IMPLIED WARRANTY, THE SCOPE AND DURATION OF SUCH WARRANTY WILL BE THE MINIMUM PERMITTED UNDER SUCH LAW. THE PRODUCTS AND SERVICES DO NOT CONSTITUTE BUSINESS OR LEGAL ADVICE. TECH LAB DOES NOT WARRANT THAT THE PRODUCTS AND SERVICES PROVIDED TO OR USED BY YOU HEREUNDER SHALL CAUSE YOU AND/OR YOUR PRODUCTS OR SERVICES TO BE IN COMPLIANCE WITH ANY APPLICABLE LAWS, REGULATIONS, OR SELF-REGULATORY FRAMEWORKS, AND YOU ARE SOLELY RESPONSIBLE FOR COMPLIANCE WITH THE SAME, INCLUDING, BUT NOT LIMITED TO, DATA PROTECTION LAWS, SUCH AS THE PERSONAL INFORMATION PROTECTION AND ELECTRONIC DOCUMENTS ACT (CANADA), THE DATA PROTECTION DIRECTIVE (EU), THE E-PRIVACY DIRECTIVE (EU), THE GENERAL DATA PROTECTION REGULATION (EU), AND THE E-PRIVACY REGULATION (EU) AS AND WHEN THEY BECOME EFFECTIVE.

Special thanks to John Clyman, VP Engineering, Marketplace Quality & Security, Magnite for his leadership.

Other Significant Contributors Include:

Paul Bannister, Chief Strategy Officer, Cafe Media; Per Bjorke, Sr. Product manager, Ad Traffic Quality, Google; Eric Bozinny, Sr. Director, Marketplace Quality, Pubmatic; Julien Delhommeau, Sr. Solutions Consultant, Xandr; Emma Fenlon, Sr. Manager, Exchange Quality, Verizon Media; Rahul Gupta, VP Client Solutions, Pulsepoint; Aaron Herrmann, Product Manager, Ads Integrity, Google; Curtis Light, Staff Software Engineer, Google; John Murphy, Chief Strategy Officer, Confiant; Alexandre Nderagakura, Technical Director, IAB Europe; Angie Pennington, Sales & Operations Strategy Lead, Verizon Media; Amit Shetty, VP Programmatic & Partnerships, IAB Tech Lab, Lindsay Superczynski-Matthies, Sr P&E Optimization Specialist, Exchange Quality, Verizon Media; Maddy Want, Director of Product, Index Exchange.

IAB Tech Lab Lead:

Amit Shetty, Vice President, Programmatic & Partnerships, IAB Tech Lab

Table of Contents

1. Background	5
Enforcement and Business Rules	6
What's Not in Scope	6
Big Picture: How buyers.json and dchain work	7
2. General Guidance	7
Publishing a buyers.json File	7
Crawling buyers.json files	10
Reading Data-Ad-Creative-Source Identifiers	10
Training ad ops staff	11
3. Guidance for Specific Types of Entities	12
DSP	12
Buyers.json Considerations	12
Examples of a DSP buyers.json	12
Initiate DemandChain Objects	13
Expose data-ad-creative-source Identifiers in Ad Markup	16
Expose Seat IDs	16
DSP with Managed Service Desk	16
SSP Acting as Intermediary or Other Intermediary	19
Buyers.json Considerations	19
Examples of an SSP buyers.json	19
Extend or Initiate DemandChain Objects	21
Agency	24
Brand	26
Publisher / App Developer	26
Ad Quality Vendor or Other Third Party	27

1. Background

As it matures, the programmatic advertising ecosystem is shifting to provide a greater degree of transparency for all participants. Beginning in 2017 with the introduction of ads.txt, and followed by the later adoption of app-ads.txt, sellers.json, and schain (SupplyChain Object), buyers now have far more visibility into the entities involved in supplying inventory than they previously did. This transparency allows them to make safer, more informed decisions about where to spend their advertising dollars. While this sell-side transparency has not, by any means, eliminated all forms of fraud and shady behavior, it has meaningfully raised the bar to help enable continued growth in programmatic transactions.

By comparison, transparency on the buy side of the ecosystem has lagged. As a result, sellers - as well as the very end users that brands hope to reach -- find themselves facing problematic, sometimes even malicious or offensive, ads that are difficult to trace and eradicate. The problem is particularly acute when it comes to malvertising (E.g. Ref: [Confiant report](#), [IAB Report](#), [Forbes article](#)).

Lack of transparency into buyer identity is one of the chief obstacles to eliminating malvertising. Malvertisers take advantage of the opaque and highly fragmented nature of the digital advertising ecosystem by exploiting the weakest entry points and jumping frequently from DSP to DSP. As soon as they are found out on one DSP, malvertisers seamlessly shift their campaigns to a new one. Publishers and SSPs often don't have insight into the identities of buyers and therefore can't block a known bad entity across all the access points at once. The end effect is whack-a-mole, with the same entity appearing again and again across different DSPs.

Often these malicious campaigns misuse branded creative assets that can reflect badly on an innocent advertiser. And repeated exposure to malware is one factor (Eg Ref: [IAB study, Report](#)) that has driven consumers' adoption of ad blockers.

The combination of buyers.json and the OpenRTB DemandChain Object, or dchain, provide significant improvements in transparency that can be used to combat malware and other problematic demand. While they do not attempt to comprehensively solve for every potential form of abuse, their widespread adoption will make abuse more difficult and harder to sustain for extended periods of time. Conversely, sellers and end users will benefit from a reduction in disruptive ads.

We also anticipate that the adoption of buy-side transparency will eventually provide a foundation for other uses that benefit the industry, such as more efficient allocation of credits for invalid traffic.

This document describes recommendations for implementing the buyers.json and DemandChain Object, and is designed to be used in conjunction with version 1.0 of the buyers.json and DemandChain Object specifications. For easy reference, it is organized by common themes such as publishing and crawling buyers.json files, and then by participant type.

Enforcement and Business Rules

Buyers.json and DemandChain Object are specifications, not certifications. While this document recommends implementation techniques and best practices, it is ultimately up to individual participants to establish their own business rules for how they apply and enforce these standards. For example, some sellers or intermediaries may choose to favor buyers that provide full transparency. Others may opt not to transact with buyers that provide insufficient transparency. Many different decisions are possible, and this document does not prescribe any particular approach - it aims only to broadly encourage transparency for the industry as a whole in order to facilitate trust, safety, and efficient transactions.

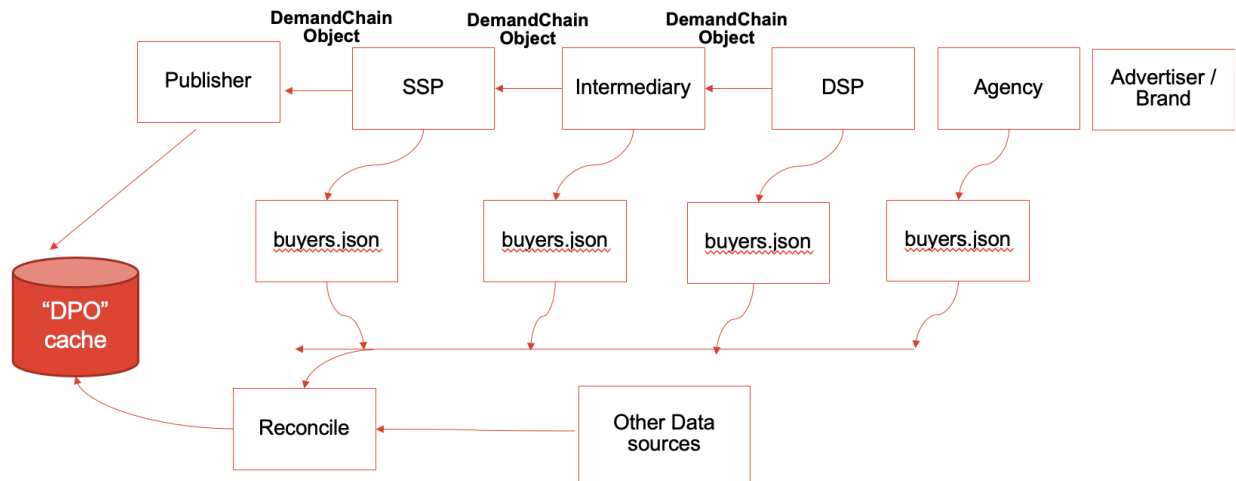
What's Not in Scope

Buyers.json and dchain don't attempt to solve every problem relating to buy-side transparency, just some of the highest-impact ones. We hope to introduce future mechanisms to provide further protection after buyers.json and dchain are widely adopted.

Buyers.json files are designed to designate the buyers' account relationships in the direction of the payor (i.e., which agencies/advertisers they are receiving demand from) and not the reverse (i.e., an entry on an agency's file does not represent a relationship with an inventory provider or demand consume such as a DSP).

In particular, these specs assume good faith on the part of participants and do not guarantee correctness of any published or transmitted data. Nor do they provide cryptographic security measures designed to ensure tamper-proofing and reduce the opportunity for creative assets to be copied and misused without appropriate authorization. Consumers should consider the possibility that the information may be misrepresented or falsified.

Big Picture: How buyers.json and dchain work



2. General Guidance

Publishing a buyers.json File

A buyers.json file (which we call a "file" even though it may be served dynamically) should be accessible via secure (HTTPS) access and should include the Content-Type: application/json HTTP header. It is recommended to also redirect from insecure URIs (i.e. <http://myspsservedomain.example/buyers.json>) for convenience of human users who may not habitually specify the https protocol in a manually-entered URI. Only a single HTTP redirect to a destination outside the original root domain is allowed. If the third-party location returns a redirect, then the advertising system should treat the response as an error.

If buyers.json contains non-ASCII characters, its contents should be encoded with UTF-8, and the Content-Type HTTP header should indicate accordingly.

The buyers.json file should be updated in a timely fashion whenever buyers that may appear in bid responses are added, and no more than one day thereafter. It should also be updated to reflect changes to existing buyer entity information. While buyer IDs are assigned on an implementer-specific basis, it is strongly recommended that implementers not reuse existing buyer seat IDs for new entities, except when the buyer seat ID represents a continuation of the business of the previous entity, such as for a mere name change or corporate merger or acquisition of the buyer. If updates to buyers.json are performed on a regular schedule, the hosting party may wish to set the HTTP Expires header on buyers.json accordingly to inform

third-party crawlers when the information will become stale and should be fetched again. HTTP Expires should not be set to more than 7 days in the future.

The buyers.json file should be constructed as valid JSON consistent with the buyers.json [specification](#), and list all entities that are able to purchase ad impressions through the platform. If the hosting party is unable or unwilling to expose the identity of specific buyer entities, it should still list all assigned buyer seat IDs, but set the is_confidential flag on any specific seat where the entity is not disclosed. Doing so allows other parties to ask for disclosure of individual confidential buyer identities via out-of-band mechanisms.

Within the buyers object array presented in the buyers.json file, the hosting party should observe the following conventions for individual fields:

Attribute	Type	Guidelines
buyer_id	String; required	Should be the ID that appears in the OpenRTB BidResponse.SeatBid.Bid.seat field, or for parties who do not pass a buyer_id, the ID that would appear in that field if supported. buyer_id should correspond to a single buyer entity. A buyer_id should not be used to aggregate multiple distinct entities unless another buyers.json file can be used to de-aggregate them (see, for example, the description of DSP with Managed Service Desk, below). It is valid for a given entity to appear with multiple buyer_id values in a buyers.json file, and encouraged to use the comments field whenever possible to differentiate the purpose of these different connections.
is_confidential	Integer; optional, default: 0	While permitted by the specification, flagging a buyer id as confidential is strongly discouraged, because it contributes to lack of transparency in the ecosystem. By way of comparison, consider that a similar is_confidential field is supported in sellers.json, but in practice is only rarely used to hide seller identities (and some DSPs will not bid on suppliers marked confidential).
buyer_type	string; required	The type of entity that is using the advertising system. If the entity controlling the buyer seat on the advertising platform is the actual brand or advertiser ultimately purchasing inventory, the buyer_type should be set to ADVERTISER. For any entity that is buying on behalf of another entity, the buyer_type value should be INTERMEDIARY; such a buyer should also post its own buyers.json file.

		<p>Entities that buy both as an end advertiser and an intermediary may be classified as BOTH and, because of their role as an intermediary, should also post their own buyers.json file like other intermediaries.</p> <p>It is recommended to include these values as all upper case.</p>
name	string; required when is_confidential = 0	The legal entity name of the buying entity. May be omitted only if is_confidential is set to 1.
domain	string; required if buyer has a web presence and is_confidential = 0	<p>The primary web domain representing the buying entity should be where the buyer posts its own buyers.json file if the buyer is an intermediary. Whenever possible, this should be the entity's corporate domain, rather than a domain representing a particular brand. This must be a domain, not a URI, and should be the root domain (Public Suffix List +1). The domain may be omitted only if is_confidential is set to 1.</p> <p>Valid example: <i>advertiser.example</i></p> <p>Invalid examples:</p> <ul style="list-style-type: none"> - <i>www.advertiser.example</i> (contains a subdomain; since .example is on the Public Suffix List, advertiser.example satisfies the Public Suffix List + 1 criterion) - <i>http://advertiser.example/index.html</i> (is a full URL rather than a domain)
comment	string; optional	May be used to further describe any entry in ways that may promote transparency or self-documentation. Its use is particularly encouraged when a buyer may be commonly known as something other than its legal entity name, or to differentiate purposes when a given buyer entity has multiple buyer IDs on the advertising system.
created_on	string; recommended	Should be included on all entries and should represent a date encoded in ISO-8601 format. This date is preferably in UTC (Zulu) time, but since the date is meant to provide a rough idea of the buyer's longevity on a platform, conversion to UTC is not strictly necessary if an implementor tracks such data in a local time zone. If the buyer seat was created after the hosting party first published its buyers.json file, the date should be the actual date of creation. If the buyer seat was created prior to the hosting party's first publishing its buyers.json file, it is recommended to include the creation date if it can determine that date with modest effort. If the

		date is very old and not easily determined with precision, for example from years ago, the hosting party may use a fixed start date of 1970-01-01 (January 1, 1970), to correspond with the Unix epoch "zero time" and convey the meaning "old and well-established".
--	--	---

Crawling buyers.json files

An ad quality vendor or other third party who is not part of the payment chain may wish to crawl buyers.json files.

Crawlers should identify themselves with an HTTP user agent documenting their operator. For example:

User-Agent: buyers.json crawler/1.0 (AmazingAQSafetyVendor, aqsafety.example)

Crawlers should be respectful of the resources of advertising systems. It is recommended to recrawl no more frequently than hourly.

Crawlers should make secure HTTPS requests to the expected location of an advertising system's buyers.json file. Normally this is expected to be located at the root domain (Public Suffix List + 1) of the DSP's advertising serving system. For example:

<https://dspssystem.example/buyers.json>

Crawlers should follow HTTP 3xx redirects. Multiple redirects are valid as long as only one redirect leads off the original root domain. Crawlers should provisionally interpret an HTTP 200 Success header as an indication that a buyers.json file has been found, but be aware that some web servers are configured to produce "friendly" page-not-found results that return an HTTP 200 status code even when the requested content does not exist. To protect against this possibility, it is recommended that crawlers look for the Content-Type: application/json header.

Crawlers should also consider the Content-Encoding header, if present, and ensure they are able to read UTF-8.

Behavior of a crawler that receives an HTTP 4xx or 5xx series response, or a buyers.json file that is not valid JSON, is implementation-dependent.

Crawlers should avoid assuming arbitrary limits on the file size or number of entities in a buyers.json file, as some advertising systems may have extremely large lists of buyers.

Reading Data-Ad-Creative-Source Identifiers

When investigating ad quality or performance issues, vendors and third parties should look for the text "data-ad-creative-source" in the markup of the creative. This string precedes identifiers that show which DSP won the impression, along with the identity of the buyer on that DSP, and the specific creative ID.

Following the data-ad-creative-source will be a comma-separated list of values. For example:
<div name="adm" data-ad-creative-source="1.0,mysp.example,ac1f9e,89708971">

The essential information about a creative and its source is contained in the string 1.0,mysp.example,ac1f9e,89708971. The "1.0" indicates that this information is compliant with buyers.json/dchain version 1.0, and is not specific to a particular creative. mydsp.example shows which DSP first inserted this creative in a bid response, and is thus the programmatic source closest to the offending buyer and creative. The following pair of identifiers indicate a specific buyer seat (ac1f93) and creative ID (89708971), respectively.

This buyer seat and creative ID can be used to notify the DSP of a problematic ad. Further, assuming the DSP supports buyers.json, the vendor can determine the identity of the specific buyer involved and correlate that activity with threat information. Take the DSP's domain (mysp.example), append "/buyers.json", and request it via HTTPS in a browser:
<https://mysp.example/buyers.json>

Training ad ops staff

If ad ops staff are fortunate enough to reproduce a problematic creative themselves, or receive a report that contains page source or network-traffic captures, they should look for the text data-ad-creative-source. This string "data-ad-creative-source" precedes identifiers that show which DSP won the impression, along with the identity of the buyer on that DSP, and the specific creative ID. In practice, multiple creatives may serve on a single web page, so ad ops staff are cautioned to ensure they are looking at the right creative; how to find the right creative on a complex web page is beyond the scope of this guide.

Following the data-ad-creative-source will be a comma-separated list of values. For example:
<div name="adm" data-ad-creative-source="1.0,mysp.example,ac1f9e,89708971">

The essential information about a creative and its source is contained in the string 1.0,mysp.example,ac1f9e,89708971. The "1.0" indicates that this information is compliant with buyers.json/dchain version 1.0, and is not specific to a particular creative. mydsp.example shows which DSP first inserted this creative in a bid response, and is thus the programmatic source closest to the offending buyer and creative. The following pair of identifiers indicate a specific buyer seat (ac1f93) and creative ID (89708971), respectively.

This buyer seat and creative ID can be used to notify the DSP of a problematic ad. Further, assuming the DSP supports buyers.json, you can determine the identity of the specific buyer involved. Take the DSP's domain (mysp.example), append "/buyers.json", and request it via HTTPS in a browser:
<https://mysp.example/buyers.json>

3. Guidance for Specific Types of Entities

DSP

A DSP or similar advertising platform that is the first programmatic access point for buyers should:

1. Publish and maintain a buyers.json file on its advertising system domain.
2. Initiate DemandChain Objects, complete ones when possible, and include them in all bid responses.
3. Expose data-ad-creative-source identifiers in ad markup in all bid responses.
4. Expose buyer seat IDs in all bid responses, if it does not already.

Buyers.json Considerations

A DSP should publish a valid buyers.json on the primary domain (Public Suffix List + 1) of its primary advertising system domain:

`https://myspserverdomain.example/buyers.json`

If a buyer_id represents multiple buyer entities, it should likely be treated as a "DSP with Managed Service Desk", which should in turn publish its own buyers.json file as described below.

Examples of a DSP buyers.json

Lines prefixed with > represent a client request from a crawler or human:

```
> GET http://myspserverdomain.example/buyers.json
HTTP/1.1 301 Moved Permanently
Location: https://www.myspserverdomain.example/buyers.json
> GET https://www.myspserverdomain.example/buyers.json
HTTP/1.1 200 Success
Content-Type: application/json; charset=utf-8
Expires: Wed, 24 Feb 2021 16:00:00 GMT
{
  "contact_email": "adquality@advertisingsystem.com",
  "contact_address": "Advertising System Inc., 101 Main Street, New
York, NY 10101",
  "version": "1.0",
  "identifiers": [
    {
      "name": "TAG-ID",
      "value": "29da45e9wbh0bd5g"
    }
  ],
  "buyers": [
```

```

{
  "buyer_id": "1234",
  "name": "Advertiser, Inc",
  "domain": "advertisingdomain.com",
  "buyer_type": "ADVERTISER",
  "created_on": "2020-10-01"
},
{
  "buyer_id": "5678",
  "name": "Intermediate Enterprises",
  "domain": "platformdomain.com",
  "buyer_type": "INTERMEDIARY",
  "created_on": "2020-09-21"
},
{
  "buyer_id": "1000023",
  "name": "Agency Group",
  "domain": "agencydomain.com",
  "buyer_type": "INTERMEDIARY",
  "comment": "Main seat for Agency Group",
  "created_on": "2020-09-01"
},
{
  "buyer_id": "212402",
  "is_confidential": 1,
  "buyer_type": "INTERMEDIARY",
  "created_on": "2020-10-01"
}
]
}

```

Initiate DemandChain Objects

As the point of entry to programmatic communication, DSPs should initiate a DemandChain Object in all bid responses. Ideally, this DemandChain Object should be complete, including all entities involved in payment for an impression up to that point.

If the entity paying for a given bid response is the ultimate payor, typically a brand or similar advertiser, the DSP should initiate a complete DemandChain Object indicating that payor. For example, the following dchain object could be inserted into the BidResponse.SeatBid.Bid.ext object:

```

"dchain" : {
  "ver": "1.0",
  "complete" : 1,

```

```

"nodes" : [
  {
    "asi": "mydspserverdomain.com",
    "bsid": "ac1f9e"
  }
]
}

```

Complete is set to 1 because the buyer here is the ultimate payor. asi is the DSP's advertising system ID (domain), also where its buyers.json can be found; bsid is the buyer seat ID purchasing the impression (if it fills). The entity associated with this buyer seat, also BidResponse.SeatBid.Bid.seat value, should be listed in the buyers.json file available at <https://mydspserverdomain.com/buyers.json>:

```

{
  "buyer_id": "ac1f9e",
  "name": "A Famous Brand",
  "domain": "afamousbrand.example",
  "buyer_type": "ADVERTISER",
  "created_on": "2020-09-01"
}

```

If the entity paying for the bid response is not the ultimate payor, but is an intermediary such as an agency, the preferred approach is for the DSP to initiate a complete DemandChain Object including the full chain of (non-programmatic) intermediaries leading to the ultimate payor, if it knows this information. For example:

```

"dchain" : {
  "ver": "1.0",
  "complete" : 1,
  "nodes" : [
    {
      "asi": null,
      "name": "A Famous Brand",
      "domain": "afamousbrand.example"
    },
    {
      "asi": null,
      "name": "An Agency",
      "domain": "anagency.example"
    },
    {
      "asi": "mydspserverdomain.com",
      "bsid": "12345"
    }
  ]
}

```

Complete is set to 1 because the DSP includes intermediaries as well as the ultimate payor. The first node, indicating that ultimate payor, has an asi of null because it is not an advertising system, and includes the entity name and its domain, but not a buyer seat ID. The second node, indicating the intermediary agency that holds a seat on the DSP, has an asi of null because it is not an advertising system and has not published a buyers.json file, and includes the entity name and its domain, but not a buyer seat ID. (If the agency is known to have published a buyers.json file with buyer IDs, the asi could be the agency's domain where the buyers.json file is found, and a bsid could be provided.) The third node, representing the DSP's participation, has an asi indicating the DSP's advertising system domain, where a buyers.json file can be found. The buyer ID at <https://myspserverdomain.com/buyers.json> would have an entry corresponding with that specific seat (An Agency) that looks like this:

```
{
  "buyer_id": "12345",
  "name": "An Agency",
  "domain": "anagency.example",
  "buyer_type": "INTERMEDIARY",
  "created_on": "2020-09-01"
}
```

If the entity paying for the bid response is not the ultimate payor, but is an intermediary including an agency, and the DSP does not know the full chain back to the ultimate payor, the DSP should initiate a DemandChain Object that is incomplete, like this:

```
"dchain" : {
  "ver": "1.0",
  "complete" : 0,
  "nodes" : [
    {
      "asi": "myspserverdomain.com",
      "bsid": "12345"
    }
  ]
}
```

In this case, the buyers.json file at <https://myspserverdomain.com/buyers.json> might include information like this:

```
{
  "buyer_id": "12345",
  "name": "An Agency",
  "domain": "anagency.example",
  "buyer_type": "INTERMEDIARY",
  "created_on": "2020-09-01"
}
```

but since the `buyer_type` is `INTERMEDIARY`, and further nodes are not provided, the `dchain` object must be marked incomplete.

Expose data-ad-creative-source Identifiers in Ad Markup

It is highly recommended that DSPs generate or pass along ad markup in the `OpenRTB BidResponse.SeatBid.Bid.adm` field, and find a way to include a `data-ad-creative-source` identifier that is appropriate for their implementation.

This serialization should include DSP domain, buyer seat ID, and creative ID as outlined in the spec [here](#).

Expose Seat IDs

OpenRTB 2.5 specifies how to expose seat IDs in bid responses, but makes doing so optional. If a DSP does not expose seat IDs in bid responses today, it is strongly encouraged to begin doing so in order to improve transparency in the ecosystem. The presence of seat IDs, particularly in conjunction with publishing a `buyers.json` file, also reduces the risk that a seller or intermediary will block a DSP's demand entirely in response to a malware outbreak.

DSP with Managed Service Desk

A DSP with a managed service desk or similar operation that employs a single buyer seat on behalf of multiple entities should:

1. Support `buyers.json` and DemandChain Object, as described above under "DSP".
2. Publish and maintain a `buyers.json` file on a domain representing the managed service desk.
3. Link the seat ID used by the managed service desk in the DSP's `buyers.json` file to the `buyers.json` file published by the managed service desk.

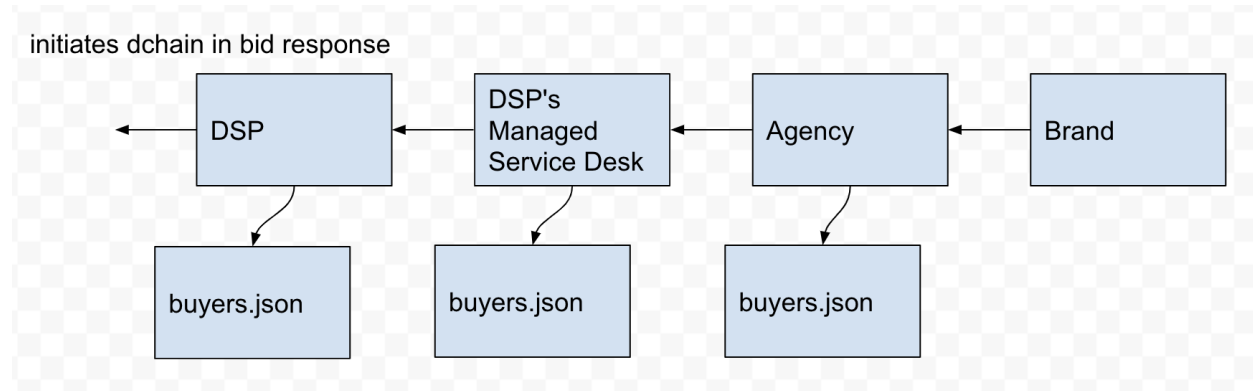
DSP support for `buyers.json` and DemandChain Object is described above. One of the entries in the DSP's `buyers.json` file should serve as a pointer to the DSP's own managed service desk and its corresponding `buyers.json`. For example, see the bold item here:

```
{
  "contact_email": "adquality@advertisingsystem.com",
  "contact_address": "Advertising System Inc., 101 Main Street, New
York, NY 10101",
  "version": "1.0",
  "identifiers": [
    {
      "name": "TAG-ID",
      "value": "29da45e9wbh0bd5g"
    }
  ],
  "buyers": [
```



```
{
  "buyer_id": "0",
  "name": "MyDSP",
  "domain": "managementservices.mydsp.example",
  "buyer_type": "INTERMEDIARY",
  "created_on": "1970-01-01"
},
{
  "buyer_id": "5678",
  "name": "Intermediate Enterprises",
  "domain": "platformdomain.com",
  "buyer_type": "INTERMEDIARY",
  "created_on": "2020-09-21"
},
{
  "buyer_id": "1000023",
  "name": "Agency Group",
  "domain": "agencydomain.com",
  "buyer_type": "INTERMEDIARY",
  "comment": "Main seat for Agency Group",
  "created_on": "2020-09-01"
},
{
  "buyer_id": "212402",
  "is_confidential": 1,
  "buyer_type": "INTERMEDIARY",
  "created_on": "2020-10-01"
}
]
```

From the perspective of buyers.json and DemandChain, a managed service desk is another intermediary that sits between the DSP platform and buyers such as agencies or brands:



Thus the managed service desk should publish its own buyers.json file distinct from the DSP platform's buyers.json file, and showing all the buyers that it represents. Following the example DSP buyers.json above, the managed service desk would post its buyers.json file at <https://managementservices.mydsp.example/buyers.json>

The buyers.json file should be constructed as valid JSON consistent with the buyers.json specification, and list all buyers that may purchase through the managed service desk. If the managed service desk is unable or unwilling to expose the identity of specific buyer entities, it should still list all assigned seat IDs, but set the is_confidential flag on any specific seat where the entity is not disclosed.

When a DSP initiates dchain objects for transactions conducted via its managed service desk, it should include a DemandChainNode representing the desk:

```

"dchain" : {
  "ver": "1.0",
  "complete" : 1,
  "nodes" : [
    {
      "asi": null,
      "name": "A Famous Brand",
      "domain": "afamousbrand.example"
    },
    {
      "asi": null,
      "name": "An Agency",
      "domain": "anagency.example"
    },
    {
      "asi": "managementservice.mydsp.example",

```

```

    "bsid": "5678"
  },
  {
    "asi": "myspservedomain.com",
    "bsid": "0"
  }
]
}

```

SSP Acting as Intermediary or Other Intermediary

An SSP or other intermediary who is not the first entity to initiate a bid request should:

1. Publish a buyers.json file on its advertising system domain.
2. Append a node to DemandChain Objects received in bid responses, or initiate an incomplete DemandChain Object if one is not present.
3. Ingest buyer seat IDs from DSP or Intermediary from OpenRTB bid response.
4. Crawl buyers.json files to identify declared relationships.
5. Recommend tracking and encouraging adoption of the buyers.json and DemandChain Object specs.
6. Consider building a filter feature for publishers and/or SSPs to verify identifiers within the buyers.json files or where no file exists at all.

(Note that participation in a header bidding integration alone, where there are no decisions made by the SSP on which demand is passed, does not constitute being an intermediary)

Buyers.json Considerations

Even though we may typically think of an SSP as representing sellers rather than buyers, demand does transact through SSP platforms. Thus an SSP, or other intermediary that is not the first entity to initiate a bid request, should publish a valid buyers.json on the primary domain (Public Suffix List + 1) of its primary advertising system domain:

<https://myspservedomain.example/buyers.json>

In most cases for SSPs, the value of the buyer_type field should be INTERMEDIARY, since SSPs normally connect to DSPs that are buying on behalf of another entity. The SSP should use a buyer_type of ADVERTISER only if the SSP is directly connected to a brand or agency in a more DSP-like relationship (in which case we recommend reviewing the DSP section above).

Examples of an SSP buyers.json

Lines prefixed with > represent a client request from a crawler or human:

```

> GET http://mysspserverdomain.example/buyers.json
HTTP/1.1 301 Moved Permanently
Location: https://www.mysspserverdomain.example/buyers.json
> GET https://www.mysspserverdomain.example/buyers.json
HTTP/1.1 200 Success
Content-Type: application/json; charset=utf-8
Expires: Wed, 24 Feb 2021 16:00:00 GMT
{
  "contact_email": "adquality@mysspserverdomain.com",
  "contact_address": "SSP Company Inc., 101 Main Street, New York, NY
10101",
  "version": "1.0",
  "updated": "2-24-2021",
  "identifiers": [
    {
      "name": "TAG-ID",
      "value": "29da45e9wbh0bd5g"
    }
  ],
  "buyers": [
    {
      "buyer_id": "4567",
      "name": "DSP, Inc",
      "domain": "dspspartnerdomain.com",
      "buyer_type": "INTERMEDIARY",
      "created_on": "2020-10-01"
    },
    {
      "buyer_id": "8910",
      "name": "Rebroadcaster DSP Co.",
      "domain": "rebroadcaster.com",
      "buyer_type": "INTERMEDIARY",
      "created_on": "2020-09-21"
    },
    {
      "buyer_id": "9101",
      "is_confidential": 1,
      "buyer_type": "INTERMEDIARY",
      "created_on": "2020-10-01"
    }
  ]
}

```

Extend or Initiate DemandChain Objects

When integration with a supplier permits it, an SSP should either extend an existing DemandChain object that it receives in a bid response, or initiate a new, incomplete DemandChain object if no object is present. For example, if the SSP is rebroadcasting demand to another SSP via OpenRTB before reaching the ultimate publisher, the SSP can pass the DemandChain object in its OpenRTB or equivalent response. In the future, DemandChain objects might usefully be integrated into other supply access mechanisms such as Prebid.js, Open Bidding, or TAM, if the dchain information is useful to the supplier for reporting or diagnostic purposes.

To extend an existing DemandChain object, the SSP appends a node representing itself to the DemandChain object it received in the bid response. The SSP must not alter the "complete" value.

INTERMEDIARY EXAMPLE 1:

In the case where the DSP provided this complete DemandChain in the bid response:

```
"dchain" : {
  "ver": "1.0",
  "complete" : 1,
  "nodes" : [
    {
      "asi": null,
      "name": "A Famous Brand",
      "domain": "afamousbrand.example"
    },
    {
      "asi": null,
      "name": "An Agency",
      "domain": "anagency.example"
    },
    {
      "asi": "dsppartnerdomain.com",
      "bsid": "12345"
    }
  ]
}
```

The SSP would place something like the following in its bid response:

```
"dchain" : {
  "ver": "1.0",
  "complete" : 1,
  "nodes" : [
    {
      "asi": null,
      "name": "A Famous Brand",
      "domain": "afamousbrand.example"
    },
    {
      "asi": null,
      "name": "An Agency",
      "domain": "anagency.example"
    },
    {
      "asi": "dsppartnerdomain.com",
      "bsid": "12345"
    },
    {
      "asi": mysspserverdomain.com,
      "name": "DSP, Inc.",
      "domain": "dsppartnerdomain.com",
      "bsid": "4567"
    }
  ]
}
```

INTERMEDIARY EXAMPLE 2:

In the case where the DSP provided this incomplete DemandChain in the bid response:

```
"dchain" : {
  "ver": "1.0",
  "complete" :0,
  "nodes" : [
    {
      "asi": "dsppartnerdomain.com",
      "bsid": "12345"
    }
  ]
}
```

The SSP would place something like the following in its bid response:

```
"dchain" : {
  "ver": "1.0",
  "complete" :0,
  "nodes" : [
    {
      "asi": "dsppartnerdomain.com",
      "bsid": "12345"
    },
    {
      "asi": "mysspserverdomain.com",
      "name": "DSP, Inc.",
      "domain": "dsppartnerdomain.com",
      "bsid": "4567"
    }
  ]
}
```

If the SSP does not receive a DemandChain object in the bid response, it should initiate a DemandChain, which should always be marked incomplete, since the SSP does not know the full chain of payors. For example:

```
"dchain" : {
  "ver": "1.0",
  "complete" :0,
  "nodes" : [
    {
      "asi": "mysspserverdomain.com",
      "name": "DSP, Inc.",
      "domain": "dsppartnerdomain.com",
      "bsid": "4567"
    }
  ]
}
```

An advertising system who appends a node to the DemandChain object should validate the syntactic format of the preceding node in real-time. If the preceding node passes syntactic validation successfully, the advertising system can append its new node and continue. If the preceding node fails syntactic validation, the advertising system can choose how to handle it, based on their own policy and preference. Broken syntax often indicates a systemic problem, so the errors should at minimum be logged and investigated; advertising systems could also choose to take stricter action such as disposing of the bid response altogether. If the advertising

system does process a bid response containing a syntactically invalid DemandChain object, it should not pass along that invalid object, but should instead initiate a new DemandChain object with complete set to 0.

Agency

An agency, holding company, or other entity that buys programmatic inventory but is not itself integrated programmatically should:

1. Publish a buyers.json file on its corporate domain listing the brands it represents.
2. Encourage transparency in the buyers.json files of the DSPs it buys through, and provide the corporate domain where its buyers.json file is located to those DSPs to include in their buyers.json files.

Since an agency is an intermediary between the payor for an advertisement and the publisher, it should publish a buyers.json file to encourage transparency and facilitate the timely identification and removal of abusive demand. This transparency can help publishers or intermediaries working on their behalf from taking overly broad actions in response to compromised campaigns.

An agency's buyers.json file might look like this:

```
{
  "contact_email": "adquality@anagency.example",
  "contact_address": "An Agency Inc., 101 Main Street, New York, NY
10101",
  "version": "1.0",
  "identifiers": [
    {
      "name": "TAG-ID",
      "value": "29da45e9wbh0bd5g"
    }
  ],
  "buyers": [
    {
      "buyer_id": "1234",
      "name": "Famous Advertiser, Inc",
      "domain": "famousadvertiser.example",
      "buyer_type": "ADVERTISER",
      "created_on": "2020-10-01"
    },
    {
      "buyer_id": "1235",
      "name": "Another Advertiser, LLC",
      "domain": "anotheradvertiser.example",
```



```

"buyer_type": "ADVERTISER",
"created_on": "2020-09-21"
},
{
"buyer_id": "1236",
"name": "Famous Advertiser, Inc.",
"domain": "famousadvertiser.example",
"buyer_type": "ADVERTISER",
"comment": "European division of Famous Advertiser",
"created_on": "2020-09-01"
},
{
"buyer_id": "1237",
"is_confidential": 1,
"buyer_type": "ADVERTISER",
"created_on": "2020-10-01"
}
]
}

```

The purpose of buyers.json and DemandChain Object is to transparently disclose all intermediaries between the publisher or content developer and the ultimate payor for an advertisement. Thus agencies, as authorized representatives of a brand, are expected to post a buyers.json file that discloses brand relationships. Agencies should also encourage the DSPs that they work with transparently disclose the agency-DSP relationship in the DSP's buyers.json files and DemandChain Object, and provide a canonical domain representing the agency -- the same domain at which buyers.json should be posted.

Since an agency is likely not operating an advertising system, it would not be expected to have seat IDs that appear in OpenRTB bid responses. It is recommended to use an internal account ID, seat ID, or similar unique identifier in the buyer_id field to unambiguously identify each of the various brand relationships it may have.

The agency should provide a domain name representing the agency's primary web presence to the DSPs or other entities it buys directly through, and encourage the inclusion of that domain name transparently in each DSP's buyers.json. The agency should use the same domain for all DSPs or other intermediaries it buys directly from.

Brand

A brand or advertiser who is the ultimate payor for inventory transacted via the programmatic ecosystem should:

1. Encourage transparency on the part of agencies or DSPs it buys through, and provide a canonical domain for those agencies or DSPs to include in their published buyers.json files.

Brand assets such as creatives or logos are often misused as part of malvertising campaigns in order to give those campaigns an air of legitimacy, and this abuse can damage brand reputations. (ref [brand hijacking post](#)) While buyers.json and DemandChain Object do not solve this problem directly, their widespread adoption can help reduce the extent and duration of malvertising campaigns and other illegitimate demand, so we encourage brands to support the standards and the transparency efforts they are part of.

Since the purpose of buyers.json and DemandChain Object is to transparently disclose intermediaries between the publisher or content developer and the ultimate payor for an advertisement, brands that are the ultimate payor have no specific implementation to perform under these specifications. However, brands should encourage the agencies or DSPs that they work with to transparently disclose the relationship in those intermediaries' buyers.json files and DemandChain Objects, and provide a canonical domain representing the brand consistently to any such intermediaries they have a relationship with.

That canonical domain should generally be the brand's corporate site, or the site most broadly associated with the brand. Note that the adomain (advertiser domain) that a brand uses to represent itself in a particular programmatic creative or campaign must comply with its DSP's requirements for adomain values, which may not always correspond to the corporate domain listed in buyers.json. For example, some DSPs may prefer more specific adomain values or even full URLs, while the value in an agency or DSP's buyers.json file pointing to a brand should always be a domain only. For example, use mybrand.example or www.mybrand.example, not <https://www.mybrand.example/>.

If a brand is not the ultimate payor for an advertisement, and is acting as an intermediary for some other entity, it is effectively acting as an agency, and is expected to post a buyers.json file as described in the Agencies section, above.

Publisher / App Developer

A publisher or app developer who exposes its inventory for programmatic sale should:

1. Train ad ops staff to read data-ad-creative-source identifiers from ad markup delivered to clients and look up the corresponding entries in a buyer's buyers.json file.
2. Encourage - and require, in the longer term - adoption of buyers.json and DemandChain Object by their SSP and DSP partners.

Since buyers.json and DemandChain are focused on buy-side transparency, publishers don't strictly *need* to do anything to support them. However, we recommend that publishers encourage their SSPs to support buyers.json and DemandChain, and we also recommend that publishers train their ad ops staff how buyers.json and DemandChain can better inform their debugging and tracing of ad quality violations.

Ad Quality Vendor or Other Third Party

An ad quality vendor or other third party who is not part of the payment chain should:

1. Crawl buyers.json files to build a graph of buyer identities, populate its contact database, and inform its reputation models.
2. Read data-ad-creative-source identifiers from ad markup delivered to clients (user agents).