DOTS Data Channel

https://tools.ietf.org/html/draft-ietf-dots-data-channel

DOTS Interim Meeting, February 2018

Presenter: Mohamed Boucadair

Agenda

- Quick walk through the protocol
- Pending Issues
- Next steps

Design Walk Through

- DOTS agents use RESTCONF for:
  - Managing aliases
  - Managing filtering entries
- The design **avoids requiring modifications** to RESTCONF
- **Re-use** the following parameters as defined in the signal-channel specification:
  - cuid: clients MUST use the same 'cuid' for both signal and data channels
  - cdid
- Assume a **default direction for filtering** rules
  - DOTS client domain is the “destination”
- **Resources quota** is configured on the server to:
  - Limit the number of alias/filtering entries to be instantiated by a DOTS client/domain
  - Clients will be granted resources only if the quota is not reached

Design Walk Through

RESTCONF Constraints & DOTS Requirements

draft-ietf-dots-data-channel-12

```
+--rw dots-client* [cuid]
   +--rw cuid                    string
   +--rw client-domain-hash?     string
   +--rw alias* [alias-name]
      +--rw alias-name             string
      +--rw target-prefix*         inet:ip-prefix
      +--rw target-port-range* [lower-port upper-port]
      |  +--rw lower-port      inet:port-number
      |  +--rw upper-port      inet:port-number
      +--rw target-protocol*       uint8
      +--rw target-fqdn*           inet:domain-name
      +--rw target-uri*            inet:uri
```

This structure allows to achieve all the required DOTS operations on aliases: Create an alias, retrieve aliases of a given client, delete an alias of a client, etc.

Design Walk Through

RESTCONF Constraints & DOTS Requirements

draft-ietf-dots-data-channel-12

```
 augment /ietf-acl:access-lists/ietf-acl:acl:
    +--rw cuid                   string
    +--rw client-domain-hash?    string
    +--rw lifetime               int32
  augment /ietf-acl:access-lists/ietf-acl:acl/ietf-acl:aces
          /ietf-acl:ace/ietf-acl:actions:
    +--rw rate-limit?   decimal64
  augment /ietf-acl:access-lists/ietf-acl:acl/ietf-acl:aces
          /ietf-acl:ace/ietf-acl:matches/ietf-acl:ipv4-acl:
    +--rw fragments?   empty
  augment /ietf-acl:access-lists/ietf-acl:acl/ietf-acl:aces
          /ietf-acl:ace/ietf-acl:matches/ietf-acl:ipv6-acl:
    +--rw fragments?   empty
  augment /ietf-acl:access-lists:
    +--rw dots-acl-order
```

This one is more problematic, e.g., it does not allow:

- To retrieve an ACL of a given client
- To retrieve all ACLs of a given client
- To delete an ACL of a given client
- To delete all ACLs of a given client

Design Walk Through

RESTCONF Constraints & DOTS Requirements

draft-ietf-dots-data-channel-13

```
module: ietf-dots-data-channel
    +--rw dots-data
       +--rw dots-client* [cuid]
          +--rw cuid             string
          +--rw cdid?            string
          +--rw aliases
          |  +--rw alias* [name]
          |     +--rw name                  string
             …
          +--rw access-lists
             +--rw acl* [name]
              …
```

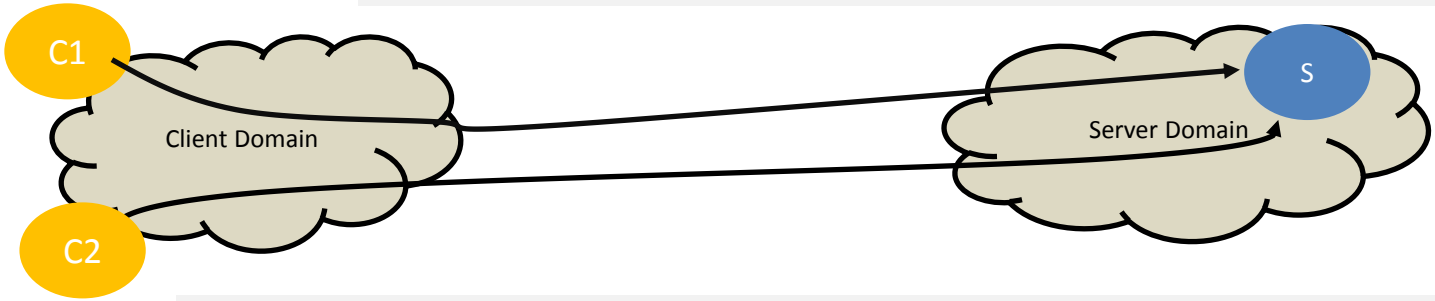
A new structure is adopted to honor DOTS requirements, while leveraging on existing specifications:

- Mimic the same structure of ACLs as defined in netmod.
- Rely upon matching criteria defined by netmod WG; a DOTS profile is defined,
- Define new actions (e.g. rate-limit)
- Support fragments

Design Walk Through

Register a DOTS Client

```
POST /restconf/data/ietf-dots-data-channel:dots-data HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
  "ietf-dots-data-channel:dots-client": [
    {
      "cuid": "dz6pHjaADkaFTbjr0JGBpw"
    }
  ]
}
```

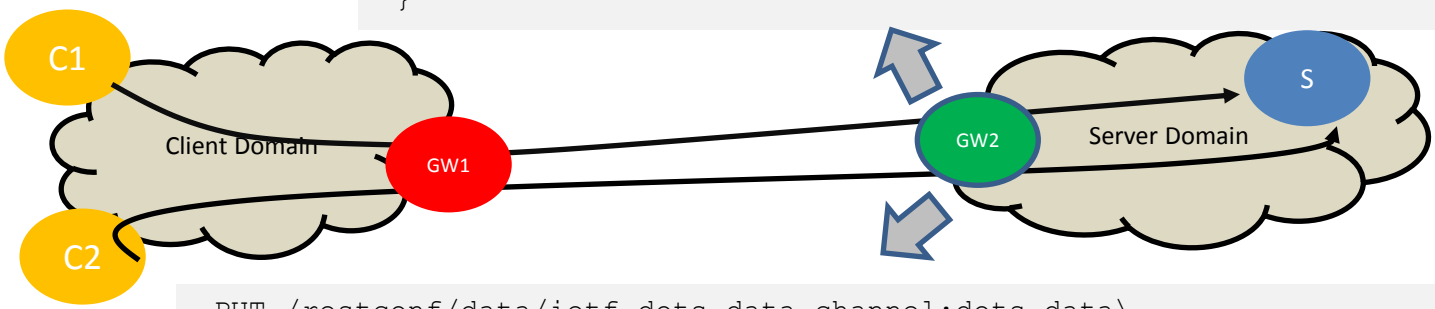



```
PUT /restconf/data/ietf-dots-data-channel:dots-data\
    /dots-client=dfrtAA78yFGhouxgioulmxw HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
  "ietf-dots-data-channel:dots-client": [
    {
      "cuid": "dfrtAA78yFGhouxgioulmxw"
    }
  ]
}
```

Design Walk Through

Server-domain Gateway addition of 'cdid' for DOTS client registration

```
POST /restconf/data/ietf-dots-data-channel:dots-data HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
  "ietf-dots-data-channel:dots-client": [
    {
      "cuid": "dz6pHjaADkaFTbjr0JGBpw",
      "cdid": "7eeaf349529eb55ed50113"
    }
  ]
}
```




```
PUT /restconf/data/ietf-dots-data-channel:dots-data\
    /dots-client=dfrtAA78yFGhouxgioulmxw HTTP/1.1
Host: {host}:{port}
Content-Type: application/yang-data+json
{
  "ietf-dots-data-channel:dots-client": [
    {
      "cuid": "dfrtAA78yFGhouxgioulmxw",
      "cdid": "7eeaf349529eb55ed50113"
    }
  ]
}
```

Design Walk Through

DOTS Operations

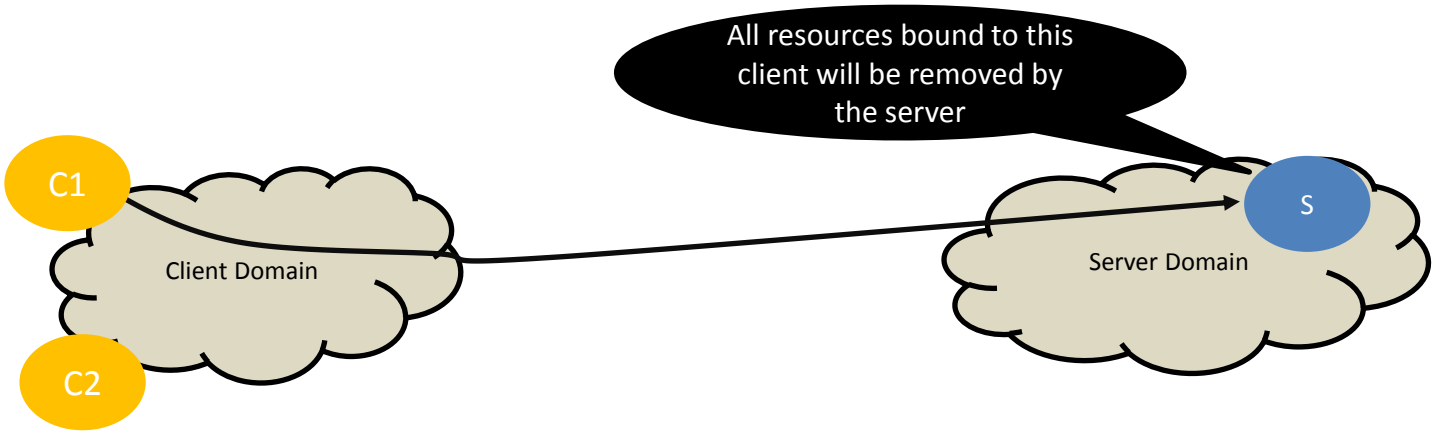
		Parameter	
	Method	**cuid**	**name**
Create an alias/filter	POST or PUT	Mandatory	Mandatory
Update an alias/filter	PUT	Mandatory	Mandatory
Delete an alias/filter	DELETE	Mandatory	Mandatory
Retrieve installed aliases/filters	GET	Mandatory	Optional

- ‘cdid’ is not required to be injected by server-side gateways once a client is registered
- It is the responsibility of the server to maintain ‘cdid’/’cuid’ associations
- DOTS servers returns “403 Forbidden” if ‘cdid’ is not present

Design Walk Through

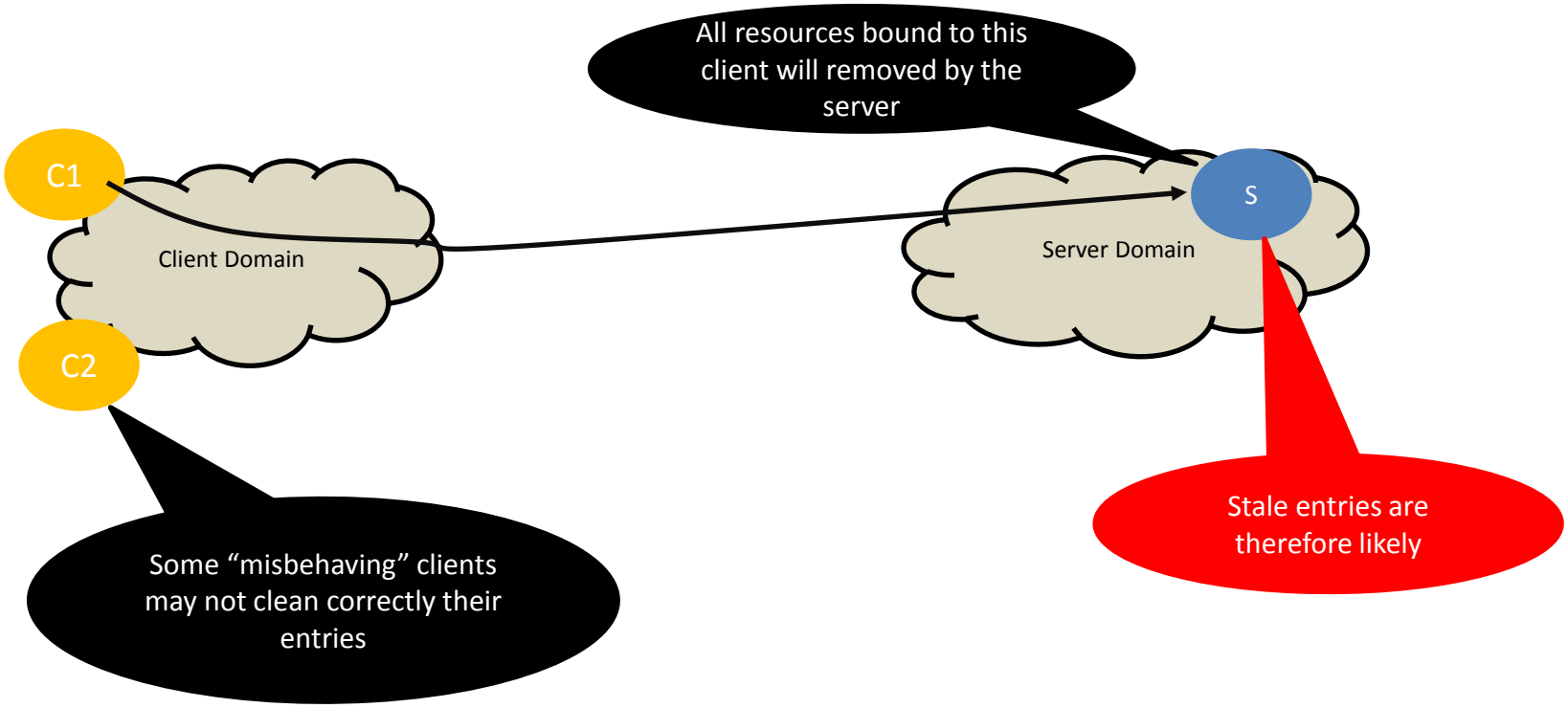
De-register a DOTS Client

```
DELETE /restconf/data/ietf-dots-data-channel:dots-data\
       /dots-client=dz6pHjaADkaFTbjr0JGBpw HTTP/1.1
Host: {host}:{port}
```

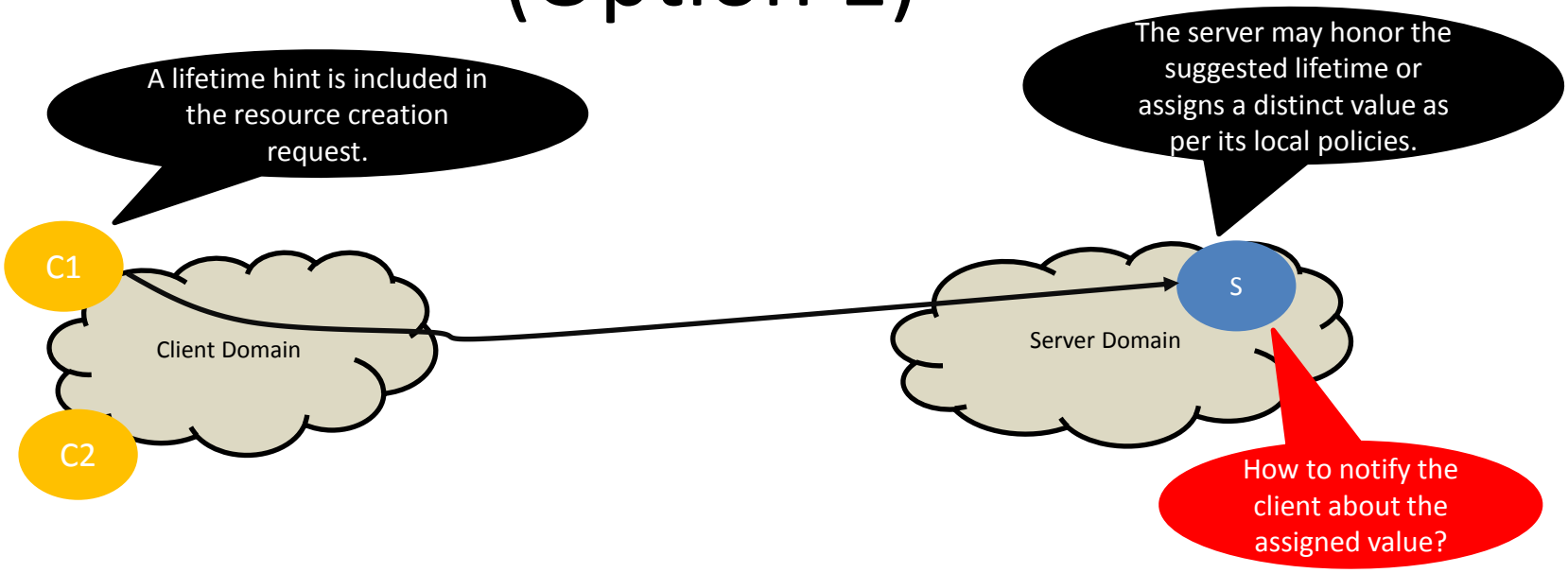


Issue #1: Avoid Stale Mappings

```
DELETE /restconf/data/ietf-dots-data-channel:dots-data\
       /dots-client=dz6pHjaADkaFTbjr0JGBpw HTTP/1.1
Host: {host}:{port}
```

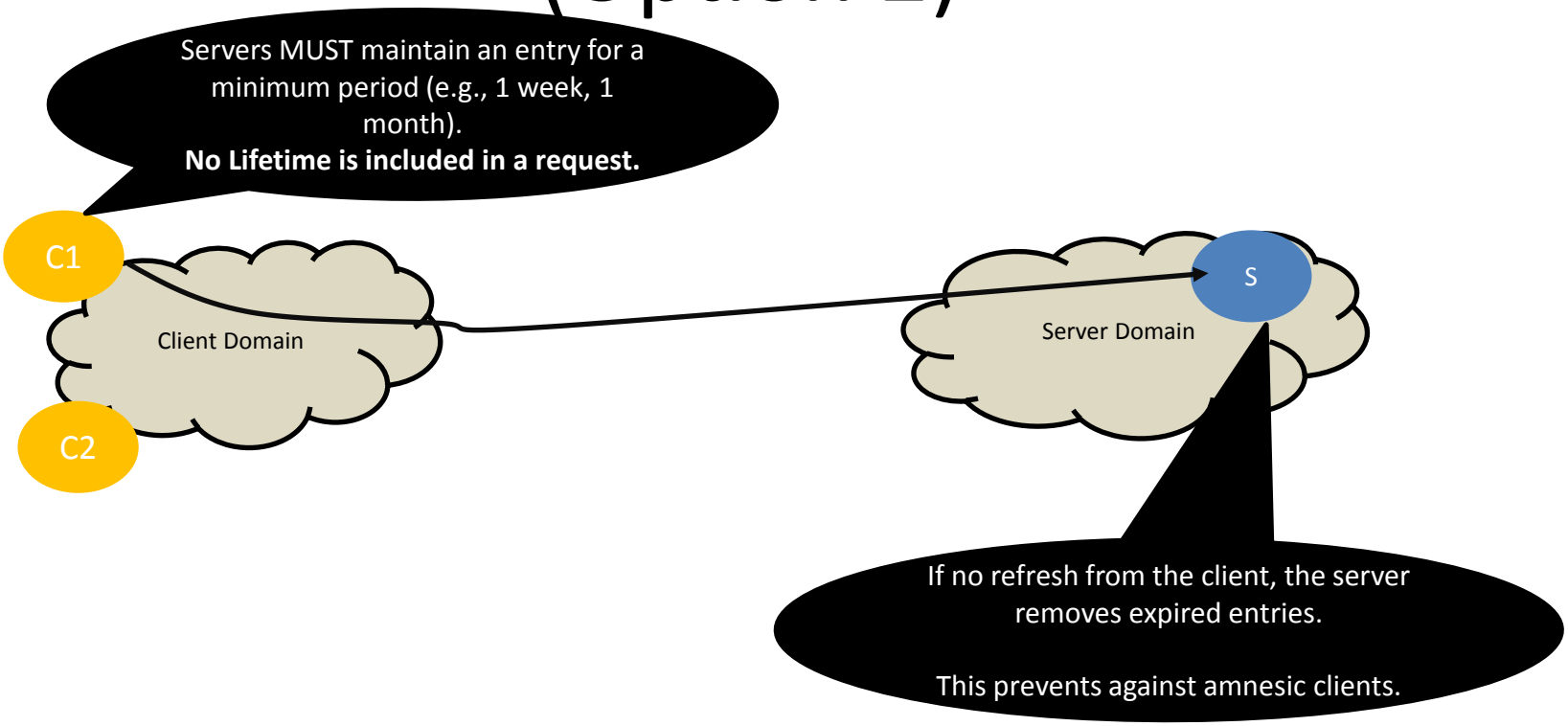


Issue #1: Avoid Stale Mappings (Option 1)



RFC8040:
"*If the POST method succeeds, a "201 Created" status-line is returned and there is* ***no response*** *message-body.*"

Issue #1: Avoid Stale Mappings (Option 2)



Suggested position: Recommended

Issue #2: Filter Activation

- Do we assume that all filtering rules are activated by default or only when a mitigation is active?
- Proposal:
  - We should ***support both***
  - The intended action is governed by a new attribute called "*activation-type*" which can be set to "immediate" or "mitigation-time"
  - Which default value to use?
- Comments?

Issue #3: Filter Direction

- Do we need to support explicit “direction” in filtering rules: “in”/“out”?
- Proposal:
  - The current default direction is aligned with the nature of DDoS attacks targeted by DOTS (incoming)
  - Direction can be defined as an extension (if needed)
  - ***No text change is required***
- Comments?

Issue #4: Global or Per-client Filters?

- Do we consider filters created by a client are globally available, or just for the client?
- Proposal:
  - ***Filters are global***
  - It is the responsibility of the client domain to ensure consistency
  - Text to be added to make it clear
- Comments?

Issue #5: Filtering Fields

- Should we supporting all of the fields as defined by “ietf-packet-fields”?
  - Do we need to define a minimum supported set?
- Proposal:
  - Adhere to what is indicated in ietf-packet-fields
  - ***No text change is needed***
- Comments?

Issue #6: Address Change

- DOTS servers MUST verify that a DOTS client is entitled to enforce filtering rules on a given IP prefix
  - What happens if that prefix is not attached to the DOTS client domain anymore (e.g., renumbering)?
- Proposal:
  - Servers MUST NOT maintain a filtering entry beyond the lifetime, unless the client asked to refresh it
  - Clients MUST update their filtering entries upon change of the destination-prefix
- Comments?

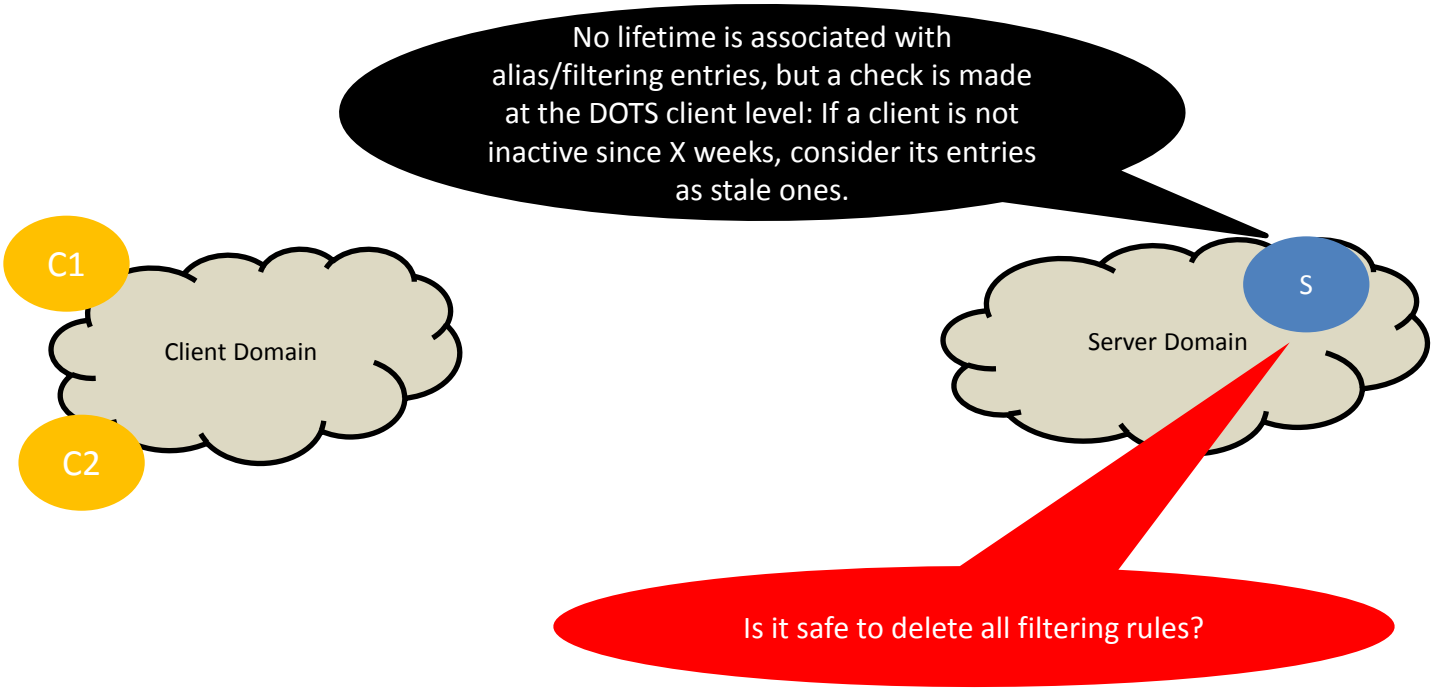
Next Steps

- Publish -14 to fix Issues #1, #2, #4, and #6
- Any issues that are not covered?
- Questions?

Appendix

Design Walk Through

Avoid Stale Mappings (option 3)



Design Walk Through

Avoid Stale Mappings (option 3, revised)