Observatory

Stephan Neuhaus, Roman Müntener, ZHAW

Elio Gubser, ETH Zurich

mami

measurement and architecture for a middleboxed internet

measurement architecture experimentation

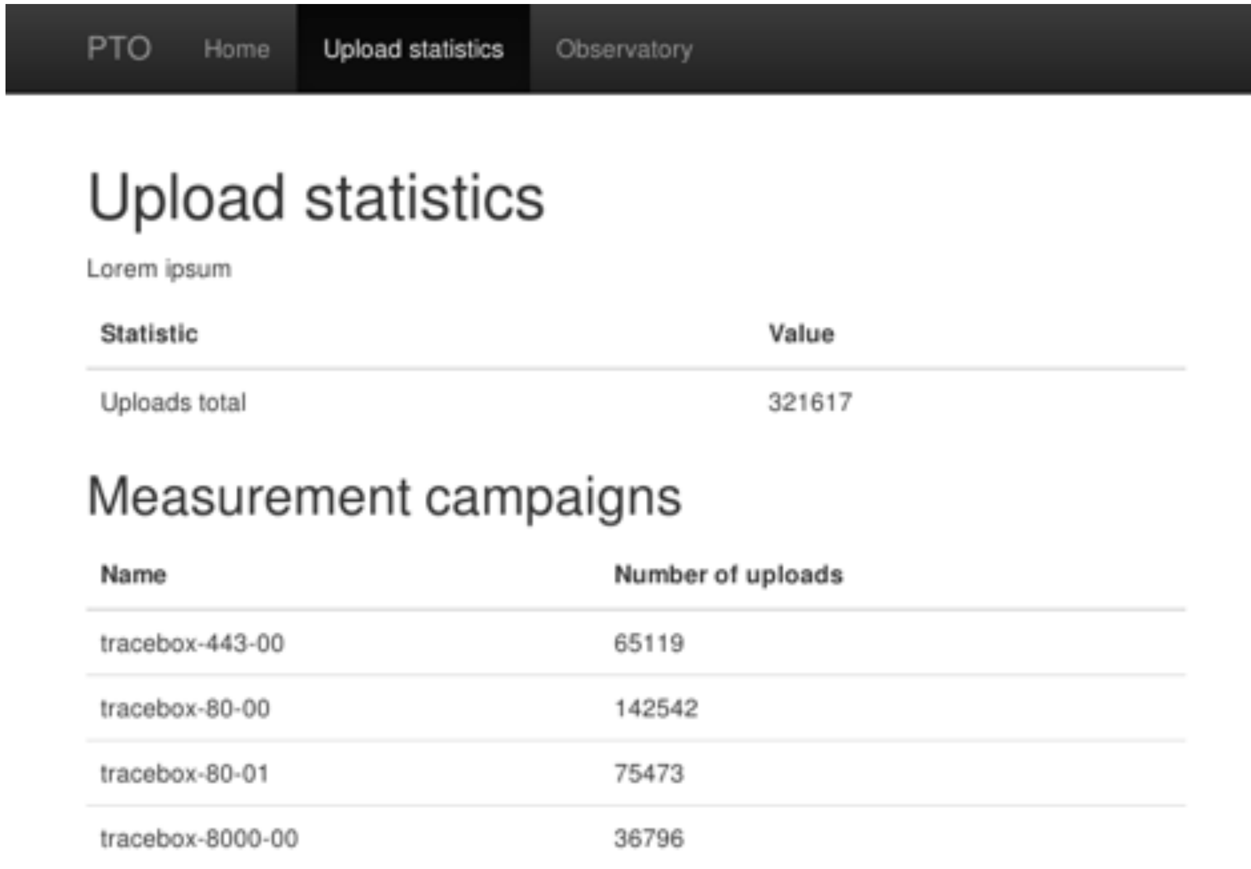
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421.The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

Data Model Overview

- Upload consists of raw data, stored in HDFS
- Raw data is fed through analysis modules
- Analysis modules produce observations, stored in Mongo
  - analysis module and raw data used
  - path and time at which the observation was made
  - conditions found to be true at the given time on the given path (e.g., `ecn.connectivity.works`)
- Potential problem: some conditions have no value: can’t distinguish between “`c` was false” and “`c` wasn’t computed”

Raw Data Management

- Upload via REST interface onto HDFS
- Upload speed now: 67 Mbit/s
  - scp about 20% faster
- Reason likely to be HDFS
- As of today, tracebox data
- REST interface documented
- Ready for production use
- (Will be closely supervised)



Upload statistics

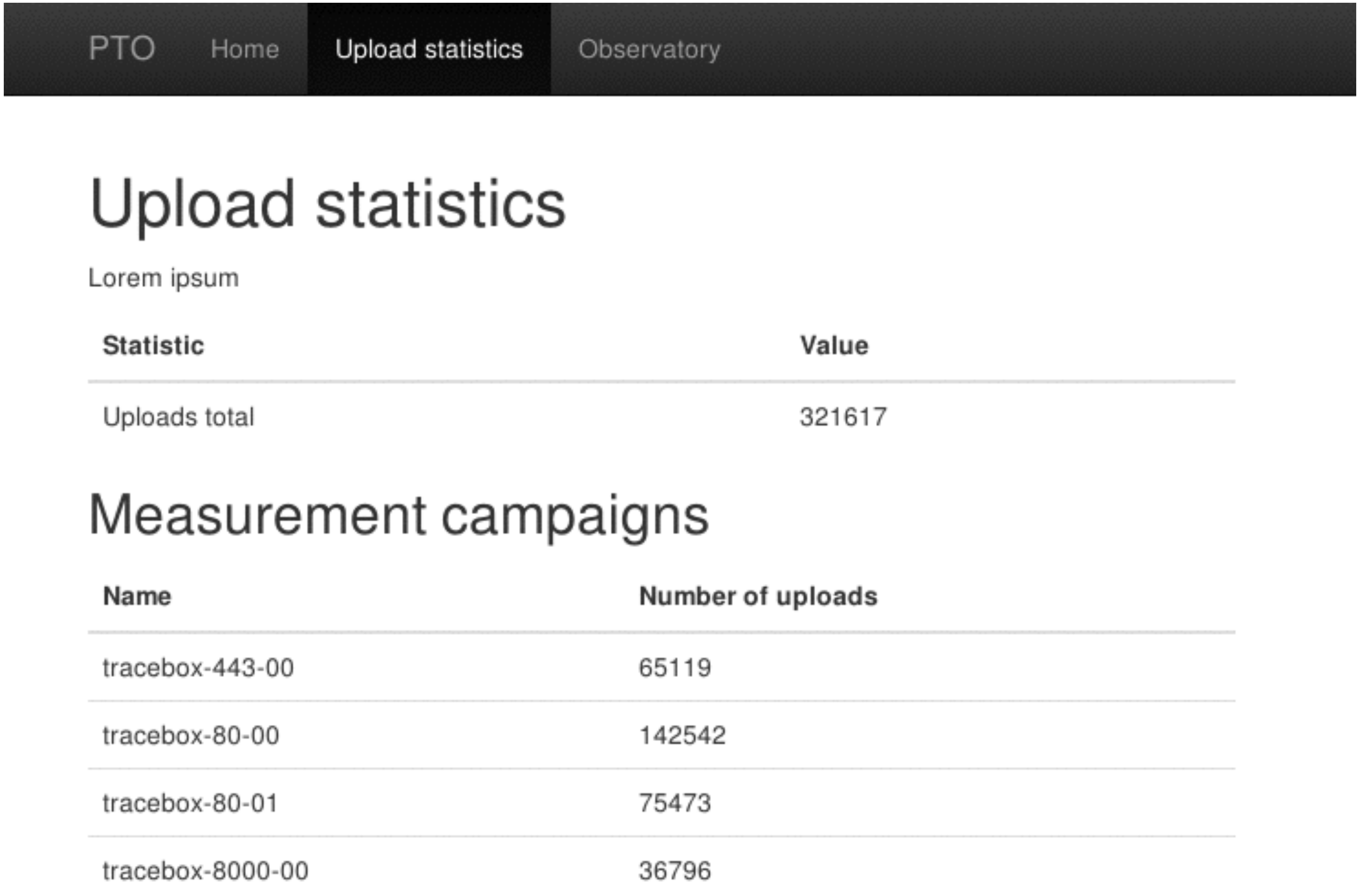
Lorem ipsum

Statistic	Value
Uploads total	321617

Measurement campaigns

Name	Number of uploads
tracebox-443-00	65119
tracebox-80-00	142542
tracebox-80-01	75473
tracebox-8000-00	36796

Raw Data Management



Upload statistics

Lorem ipsum

Statistic	Value
Uploads total	321617

Measurement campaigns

Name	Number of uploads
tracebox-443-00	65119
tracebox-80-00	142542
tracebox-80-01	75473
tracebox-8000-00	36796

Path Transparency Observatory (PTO) Core

Sample Query

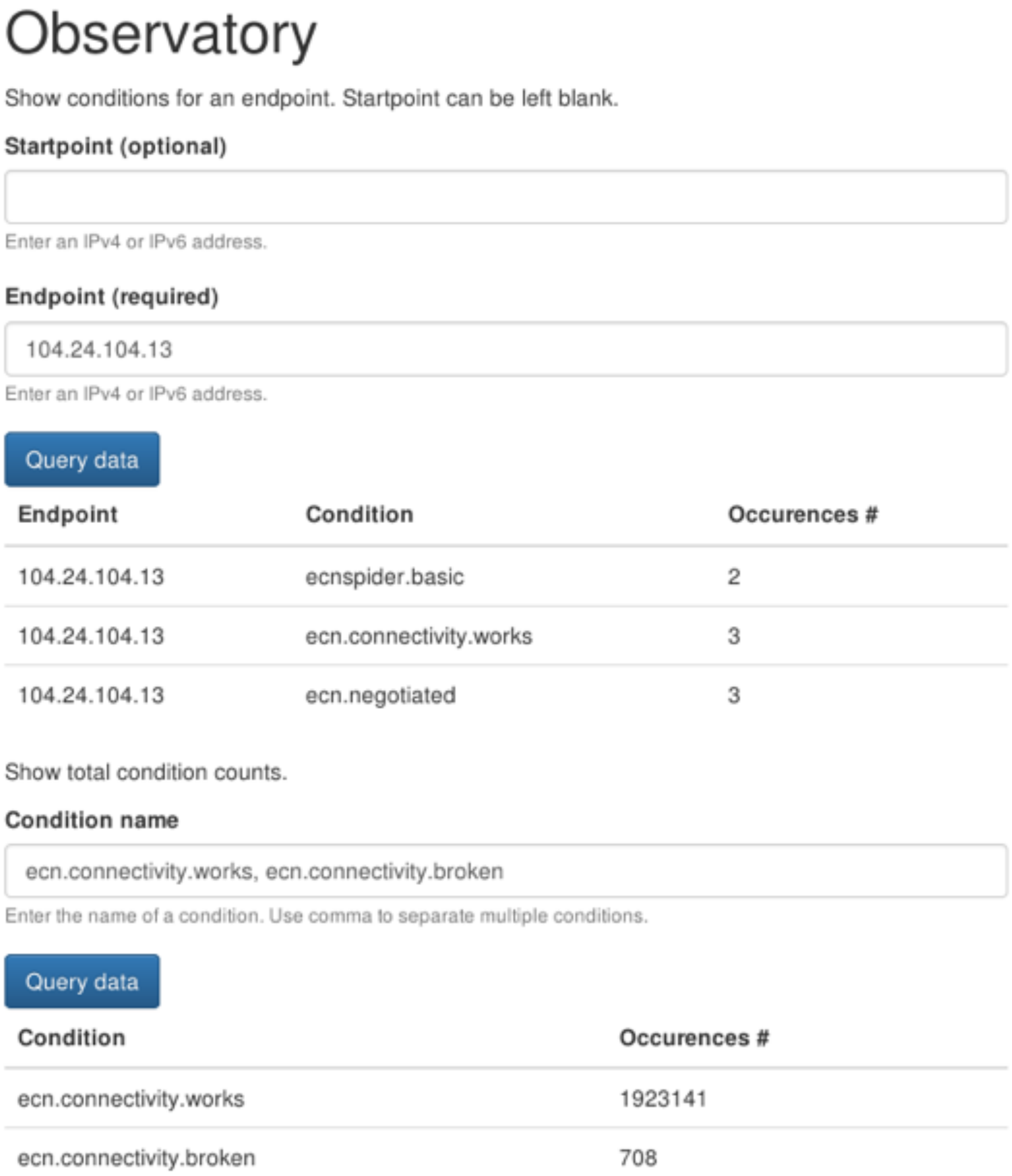
Rough SQL equivalent:

```
SELECT dip, COUNT(condition), condition
  FROM observations
  WHERE valid = True
    AND sip = $sip
    AND dip = $dip
  GROUP BY dip, condition;
```

Mongo query:

```
pipeline = [
      {'$match' : {'action_ids.0.valid' : True}},
      {'$match' : {'path' : dip}},
      {'$unwind' : '$conditions'},
      {'$project' : {'_id' : 1, 'conditions' : 1, 'sip' : { '$arrayElemAt': ['$path',0] },
                     'dip' : { '$arrayElemAt' : ['$path', -1]}}},
      {'$match' : sip_match},
      {'$match' : {'dip': dip}},
      {'$group' : {'_id' : { 'condition' : '$conditions', 'dip' : '$dip'}, 'count' : {'$sum' : 1}}},
      {'$group' : {'_id' : '$_id.dip',
                   'data' : {'$addToSet' : { 'condition' : '$_id.condition', 'count' : '$count'}}}}
    ]
```

PTO Web



Observatory

Show conditions for an endpoint. Startpoint can be left blank.

Startpoint (optional)

Enter an IPv4 or IPv6 address.

Endpoint (required)

104.24.104.13

Enter an IPv4 or IPv6 address.

Endpoint	Condition	Occurences #
104.24.104.13	ecnspider.basic	2
104.24.104.13	ecn.connectivity.works	3
104.24.104.13	ecn.negotiated	3

Show total condition counts.

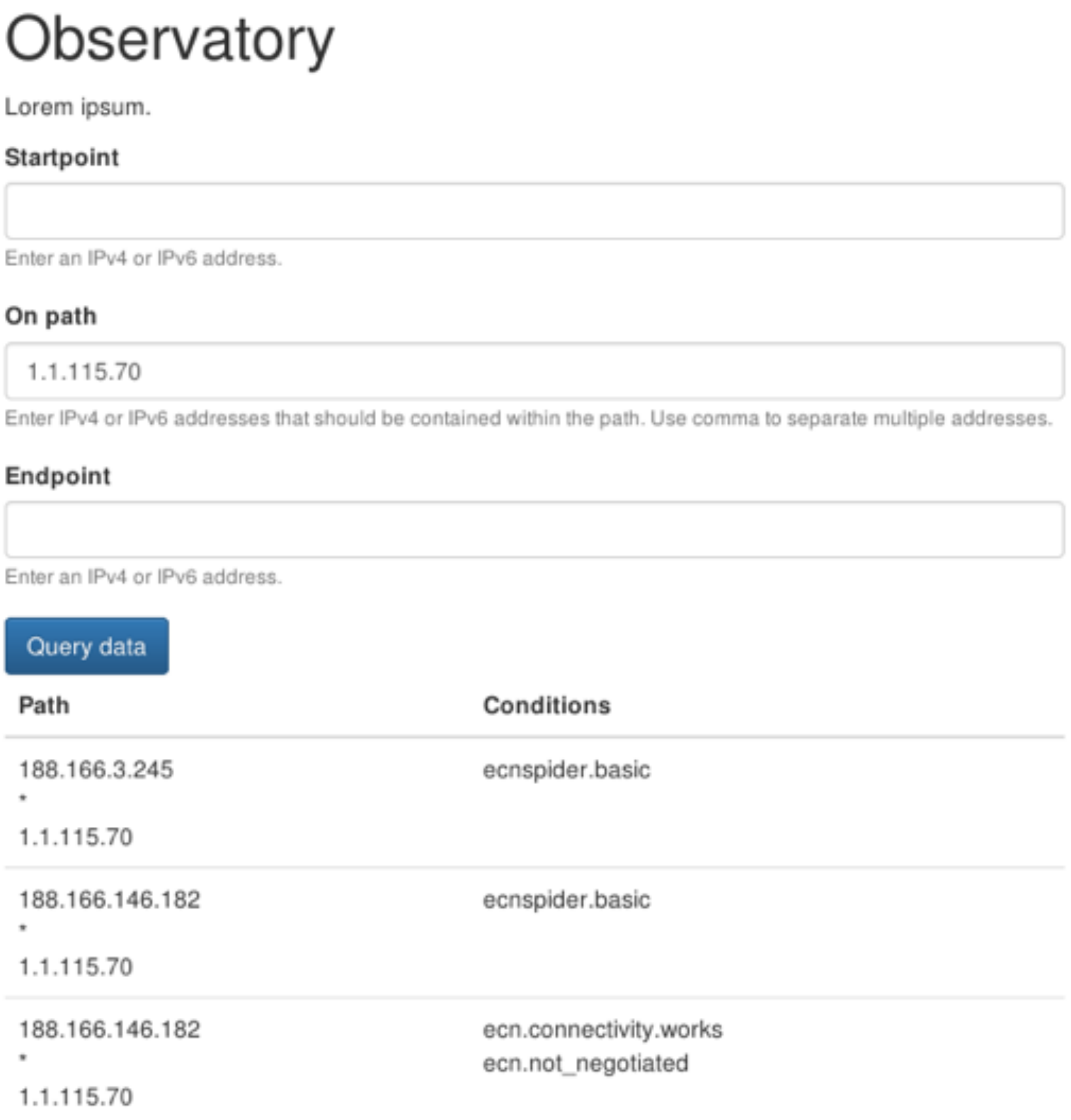
Condition name

ecn.connectivity.works, ecn.connectivity.broken

Enter the name of a condition. Use comma to separate multiple conditions.

Condition	Occurences #
ecn.connectivity.works	1923141
ecn.connectivity.broken	708

PTO Web



Observatory

Lorem ipsum.

Startpoint

Enter an IPv4 or IPv6 address.

On path

1.1.115.70

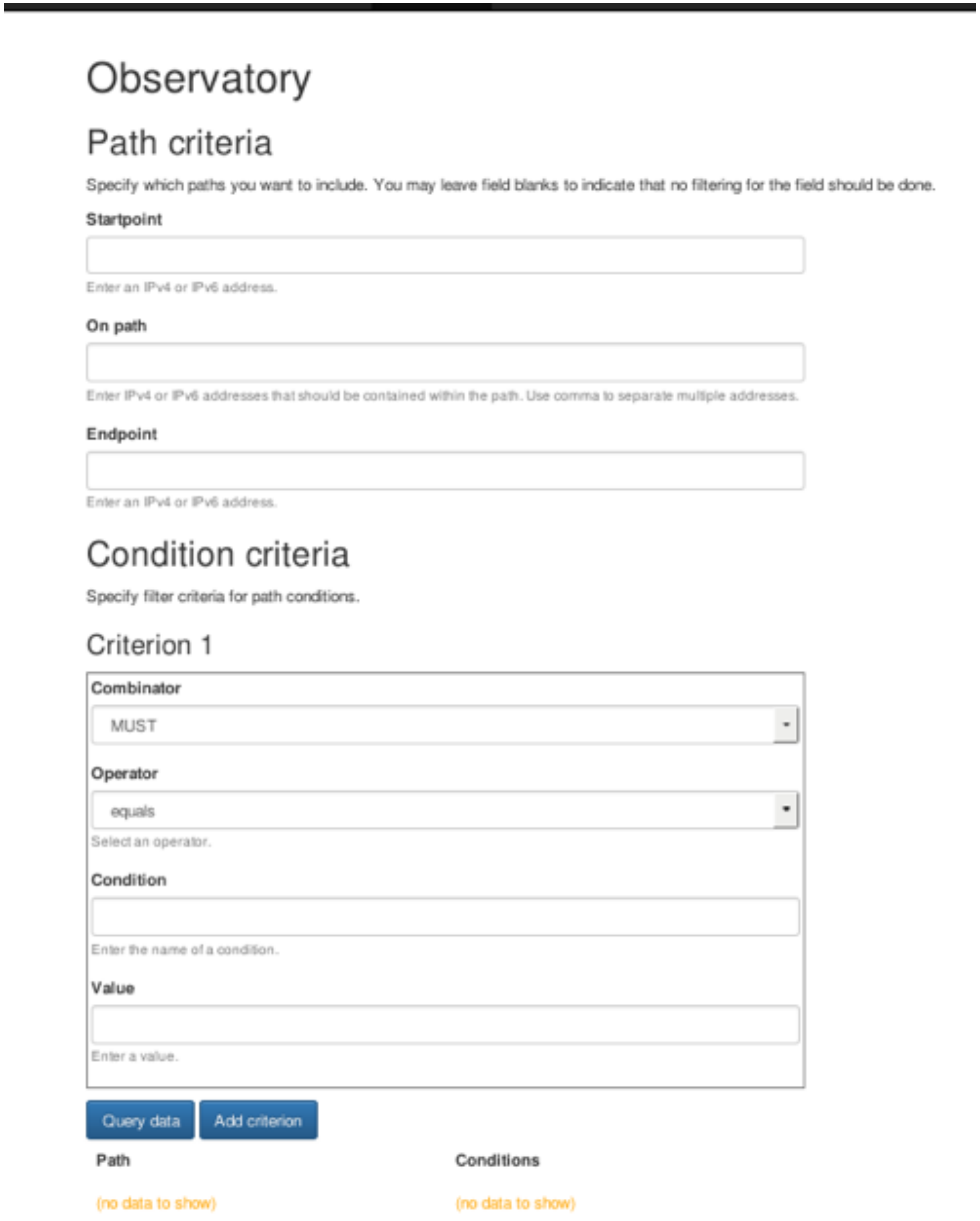
Enter IPv4 or IPv6 addresses that should be contained within the path. Use comma to separate multiple addresses.

Endpoint

Enter an IPv4 or IPv6 address.

Path	Conditions
188.166.3.245 * 1.1.115.70	ecnspider.basic
188.166.146.182 * 1.1.115.70	ecnspider.basic
188.166.146.182 * 1.1.115.70	ecn.connectivity.works ecn.not_negotiated

PTO Web (Work in Progress)



Observatory

Path criteria

Specify which paths you want to include. You may leave field blanks to indicate that no filtering for the field should be done.

Startpoint

Enter an IPv4 or IPv6 address.

On path

Enter IPv4 or IPv6 addresses that should be contained within the path. Use comma to separate multiple addresses.

Endpoint

Enter an IPv4 or IPv6 address.

Condition criteria

Specify filter criteria for path conditions.

Criterion 1

Combinator

MUST

Operator

equals

Select an operator.

Condition

Enter the name of a condition.

Value

Enter a value.

Path	Conditions
(no data to show)	(no data to show)

PTO Web: Future Work(?)

- Query language based on LISP-like SEXPs:

```
(or (== ecn.connectivity.works) (> ping 500))
```

- compile to Mongo query (relatively simple):

```
{$or : [{ecn.connectivity.works : {$eq : true}},
        {ping : {$gt : 500}}]}
```

- Would have to go through all rows (expensive)
- Optimisations are possible