ALTO Code Base and Deployments

Jordi Ros-Giralt and Kai Gao, on behalf of the ALTO Code Base Project

https://github.com/openalto

IETF Plenary 113

ALTO WG Session

3/23/2022

Table of Contents

ALTO Architecture

IETF Application Layer Traffic Optimization WG

- IETF ALTO Charter:
  - Standardizing a network API that applications can query to get the state of the network and to use this information to optimize their performance.
- IETF ALTO history of applications / use cases:
  - Peer-to-peer applications → CDNs → {Cloud, Edge-computing, 5G, V2X, XR, Science}
- Participation from a variety of carriers, vendors and universities:
  - Nokia, Ericsson, Verizon, Comcast, Telefonica, Deutsche Telekom, Huawei, China Telecom, Google, Cisco, Samsung, Qualcomm, Yale University, Tsinghua/Sichuan/Tongi Universities, etc.

IETF ALTO Architecture

https://www.rfc-editor.org/rfc/rfc7285.txt

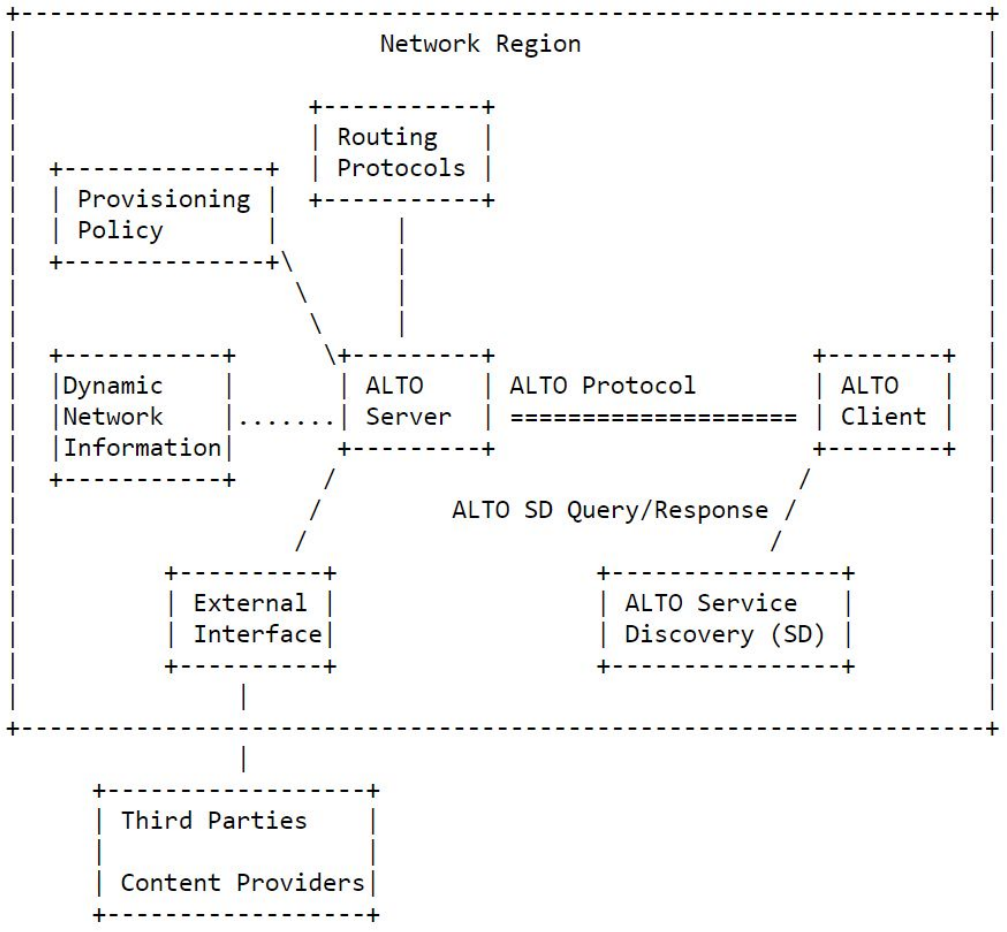


Figure 1: Basic ALTO Architecture

IETF ALTO Architecture

https://www.rfc-editor.org/rfc/rfc7285.txt

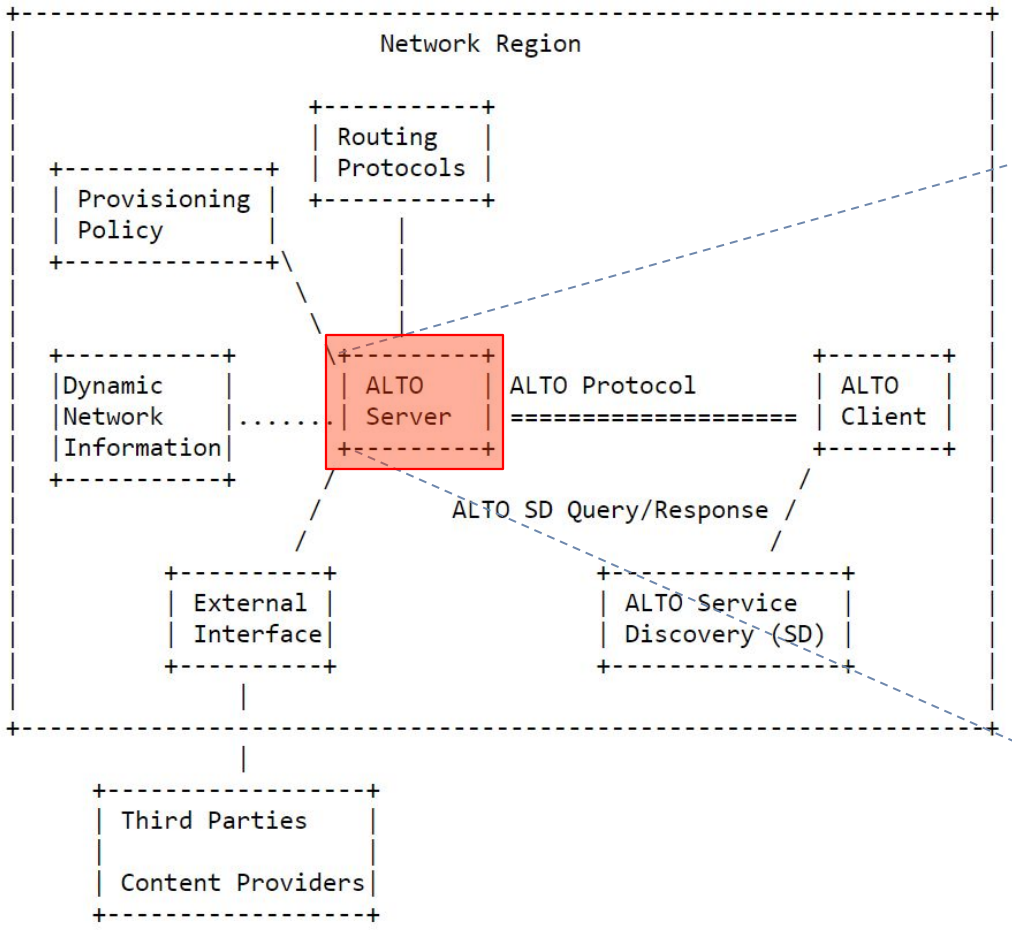
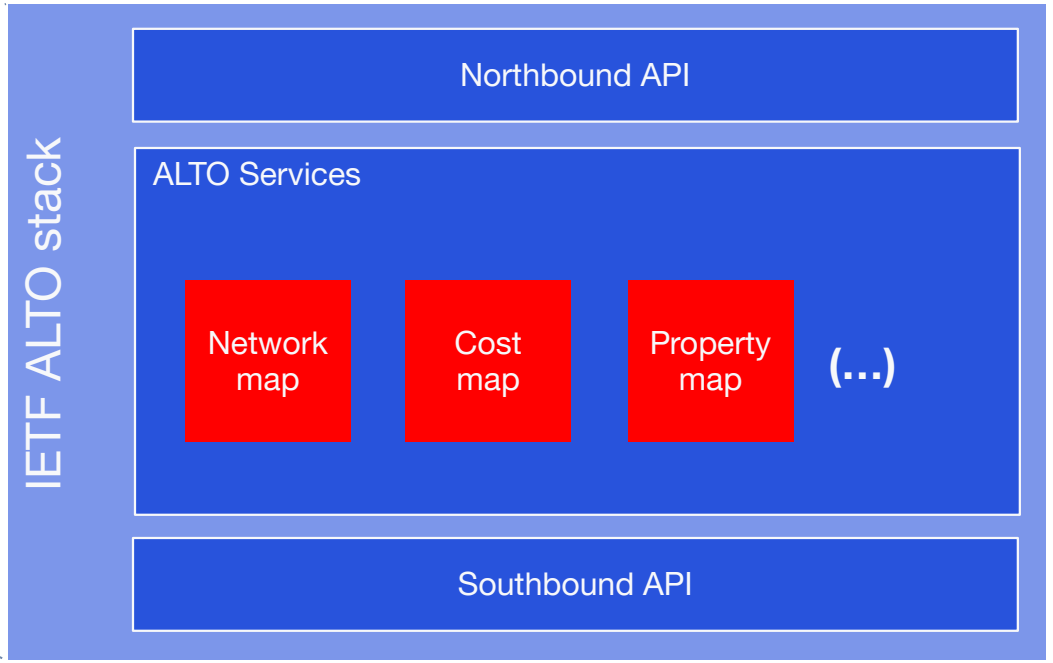


Figure 1: Basic ALTO Architecture



IETF ALTO Architecture

https://www.rfc-editor.org/rfc/rfc7285.txt

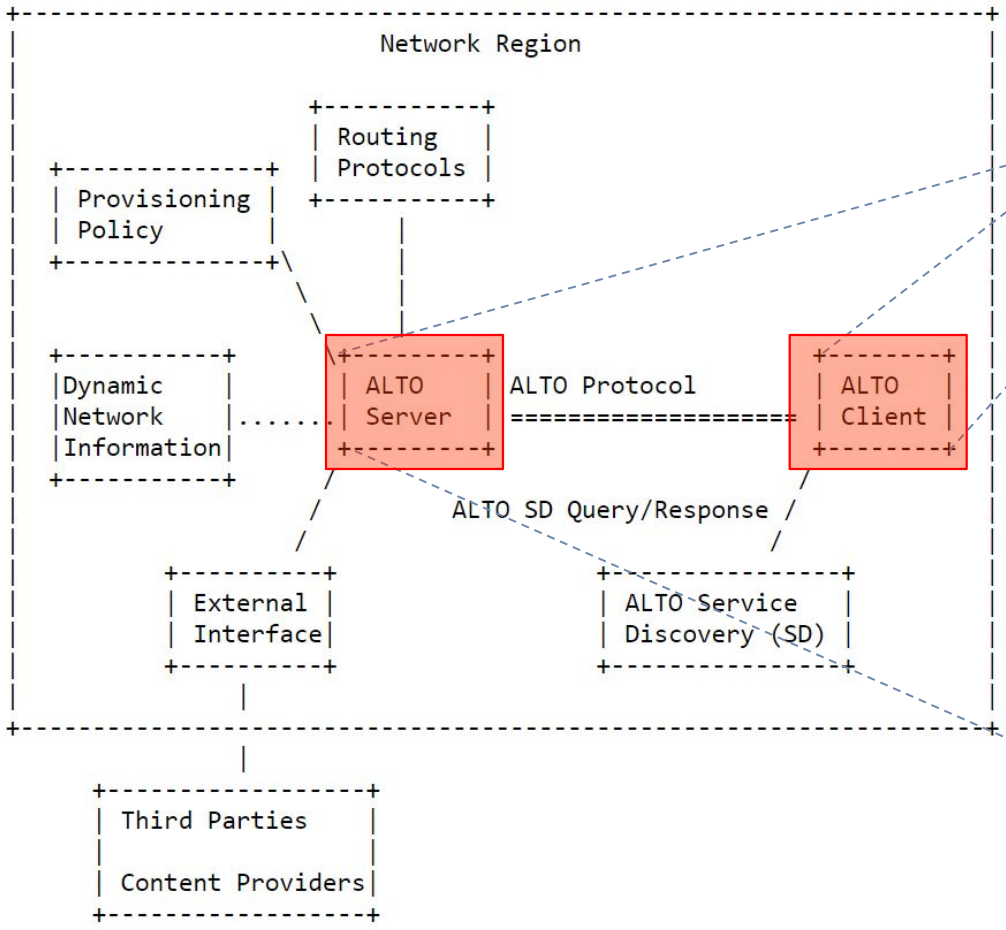
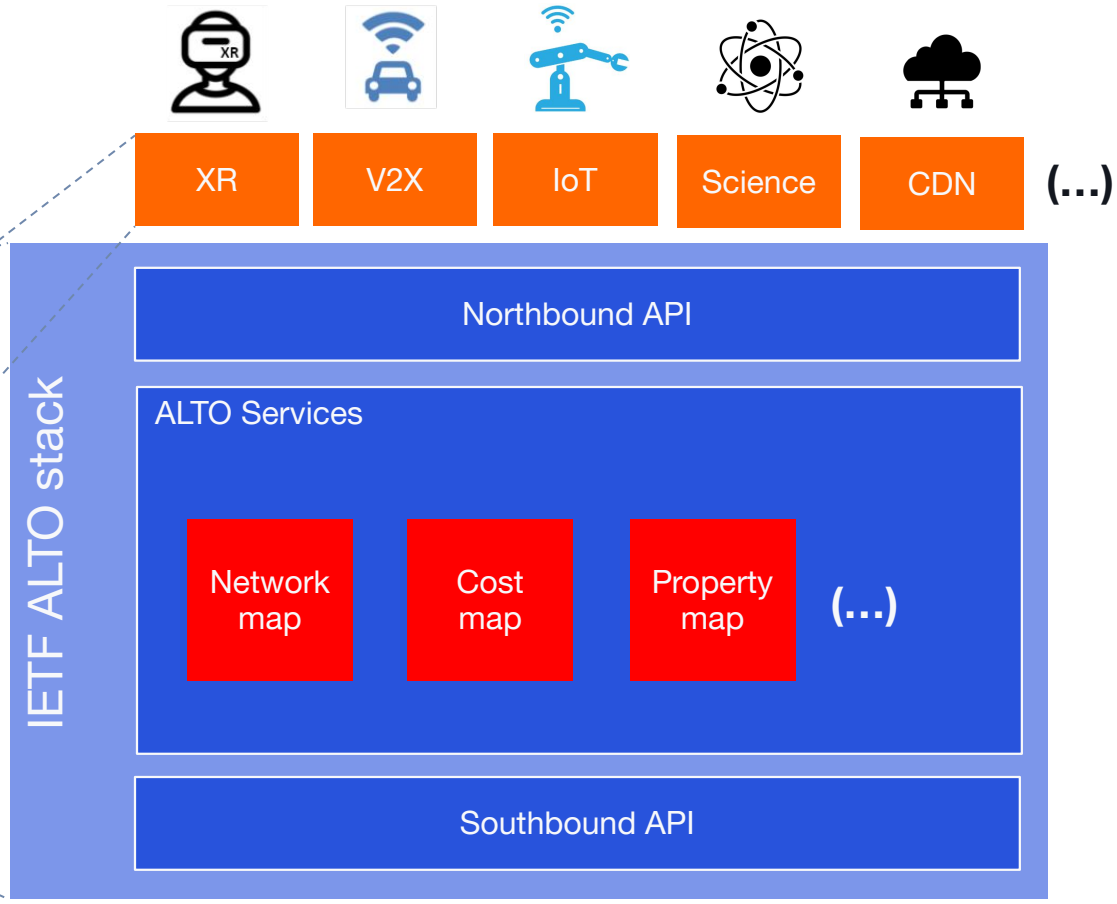


Figure 1: Basic ALTO Architecture



IETF ALTO Architecture

https://www.rfc-editor.org/rfc/rfc7285.txt

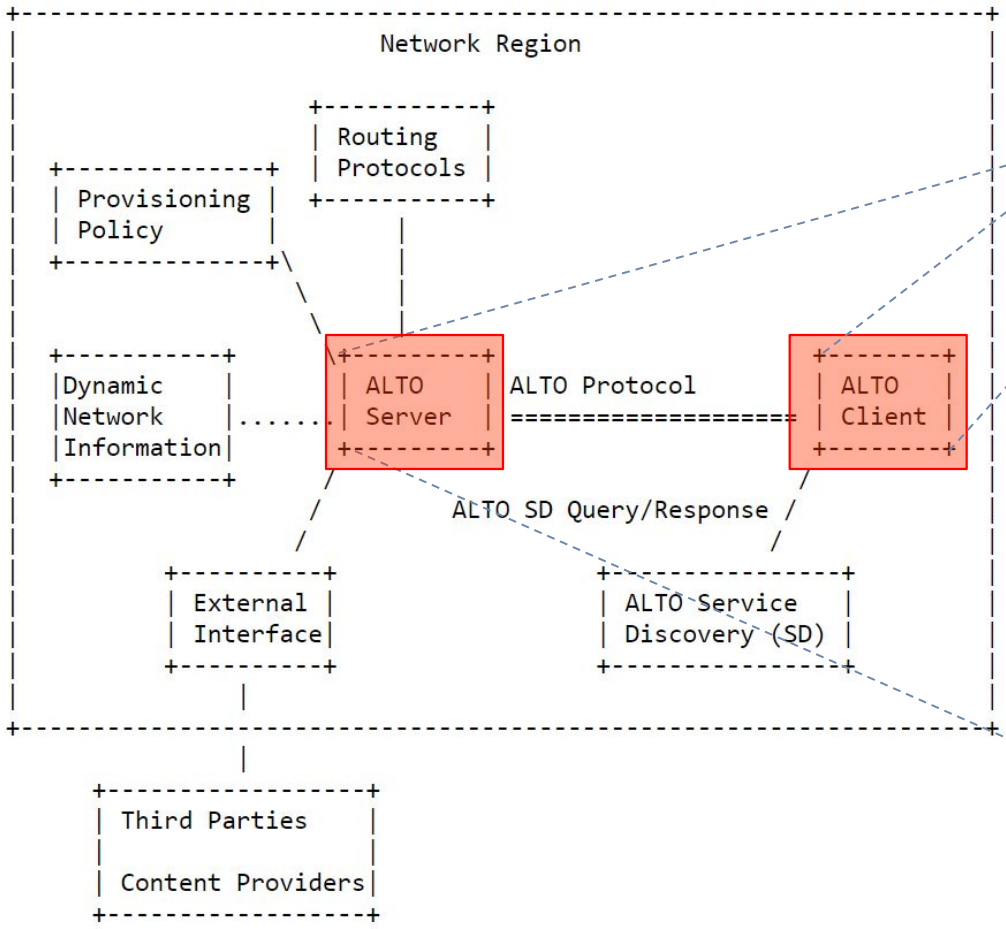
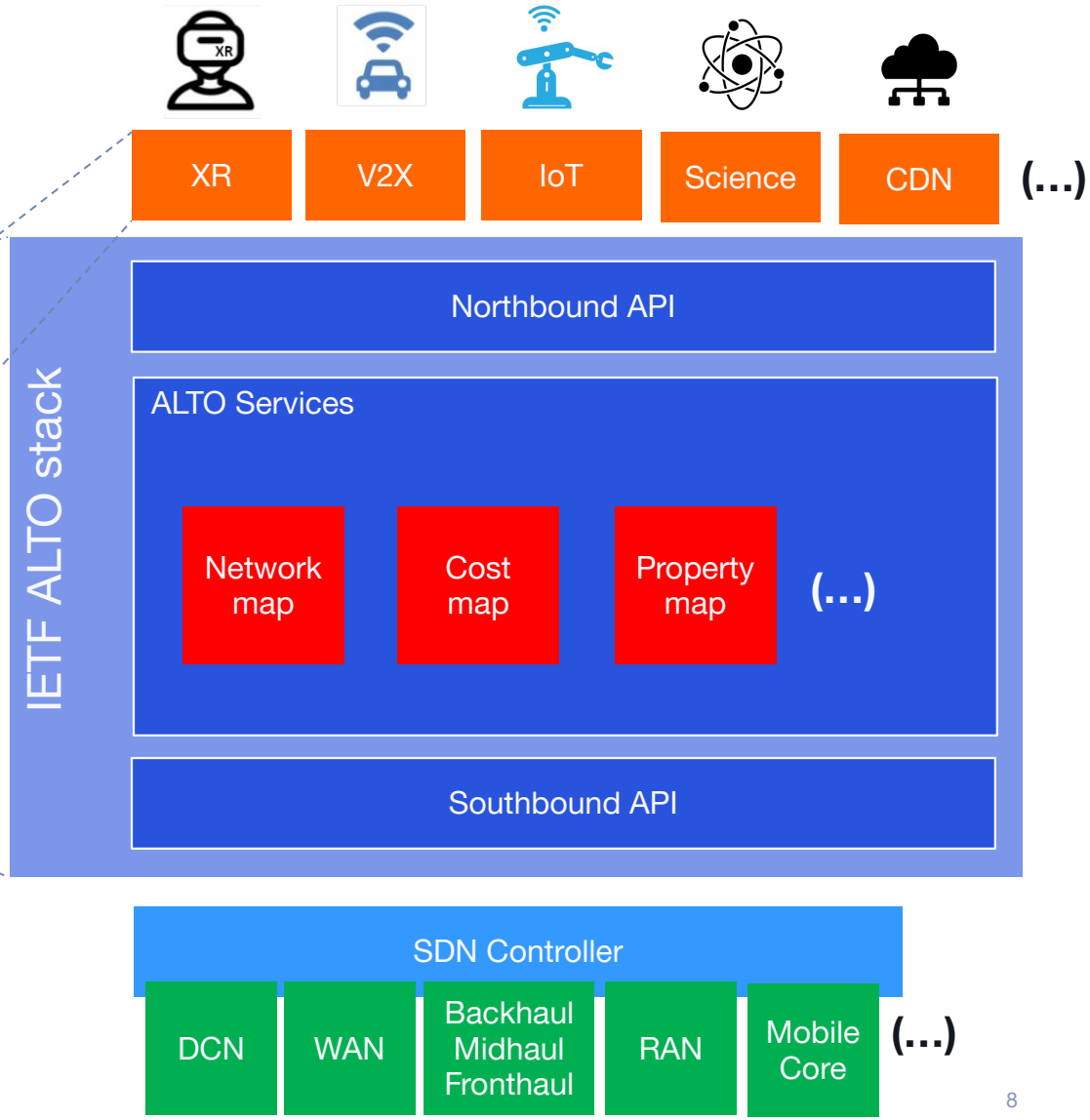
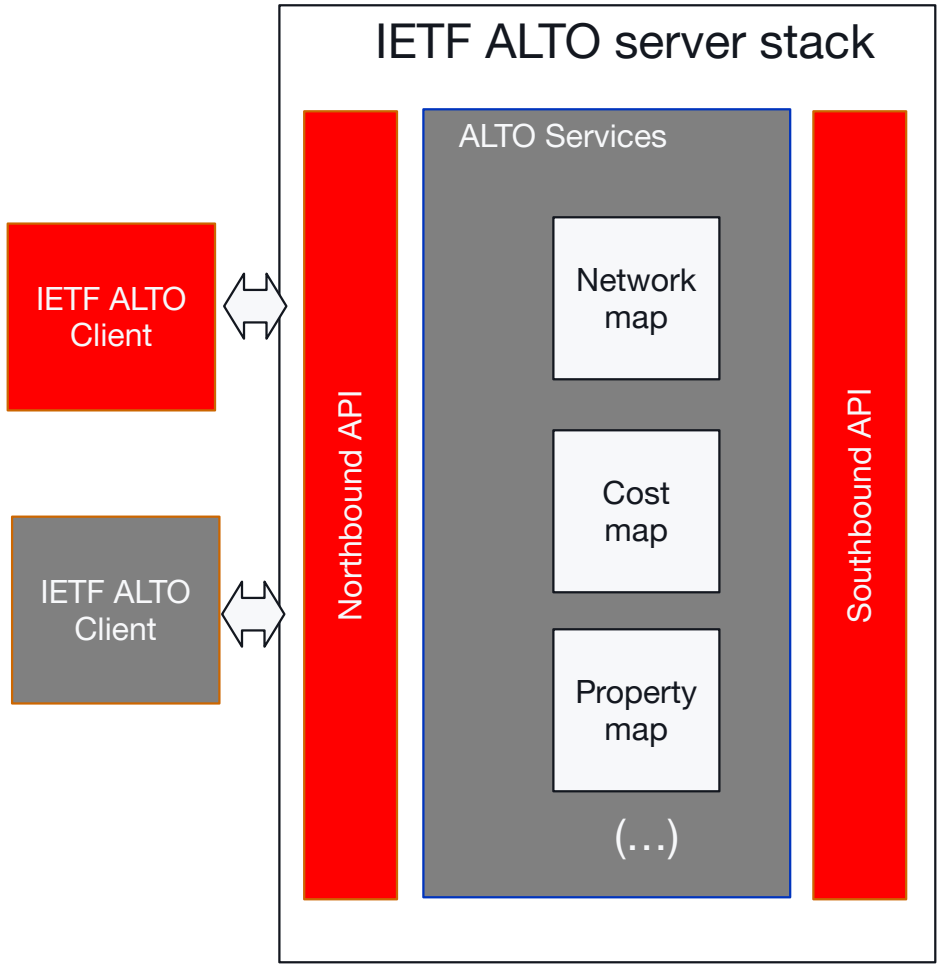


Figure 1: Basic ALTO Architecture



ALTO Code Base Project

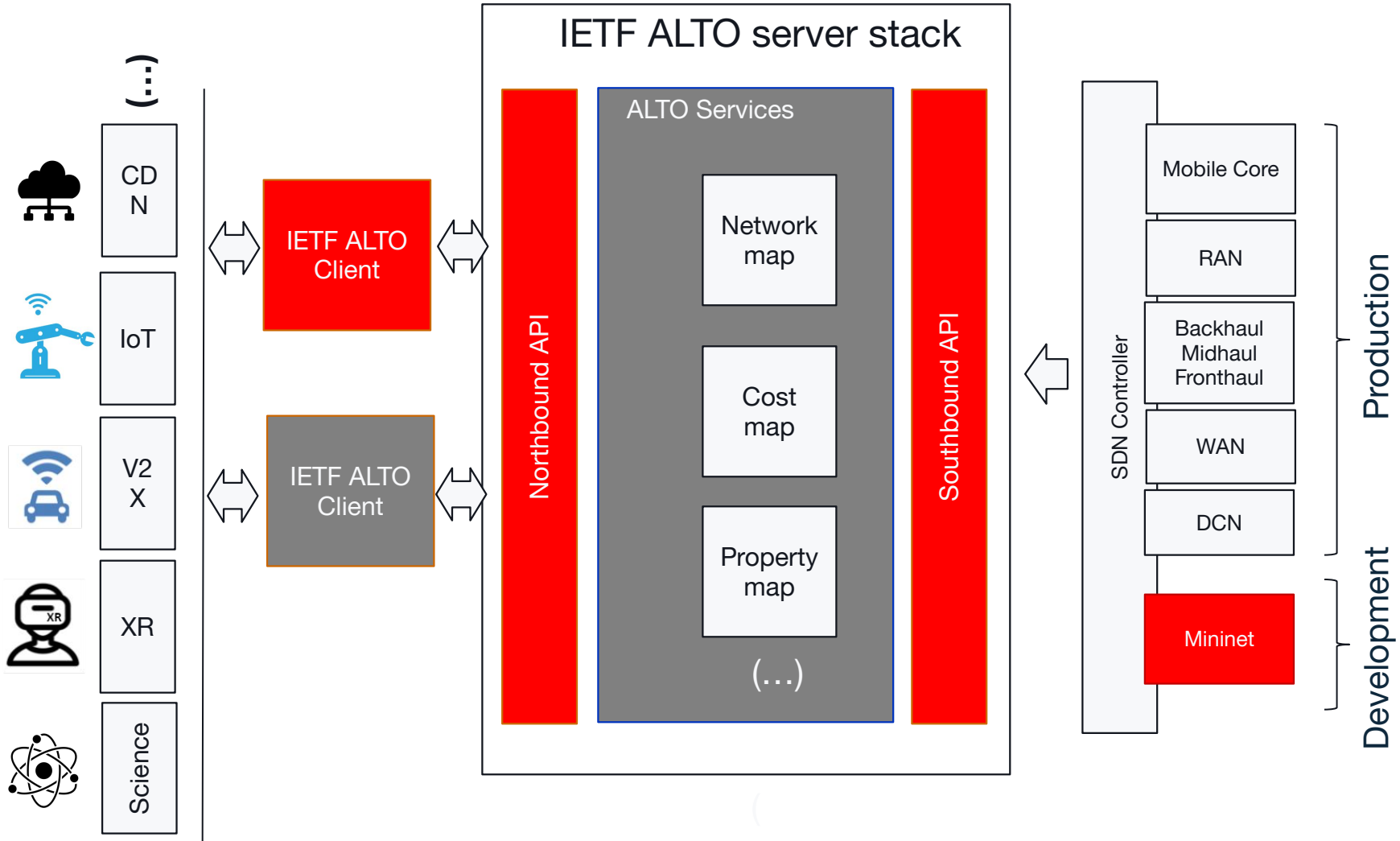
IETF ALTO Code Base



Open-source vendor independent code

Open-source vendor independent and close-source vendor specific code

IETF ALTO Code Base



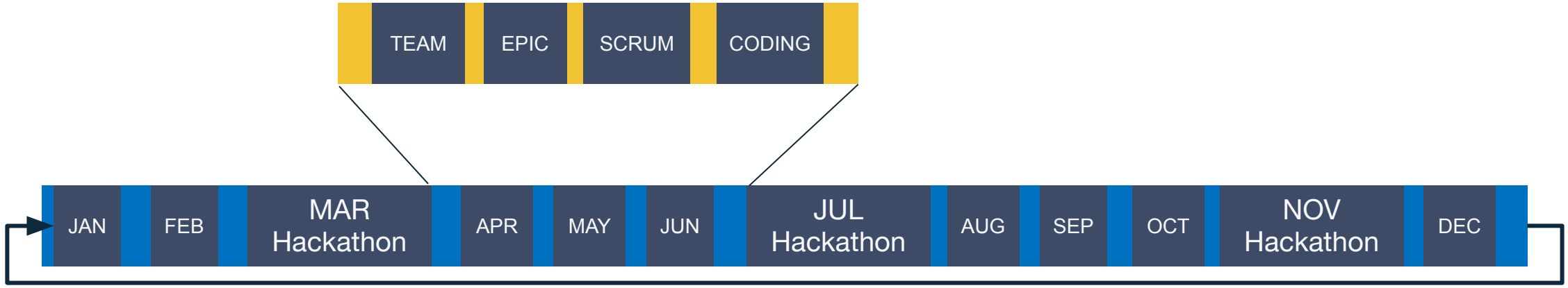
Open-source vendor independent code

Open-source vendor independent and close source vendor specific code

Project Management and Approach

Project Management and Approach

- The ALTO Code Base Project aims at providing a parallel track to the WG's standardization effort towards implementing the features introduced in the latest RFCs.
- IETF Hackathons will be used as 3-checkpoints a year to test interoperability, demo latest standard capabilities and identify issues and improvements for standardization.
- Identify and build production, open-source environments for use cases and deployment ("lean startup") to help steer ALTO standardization.

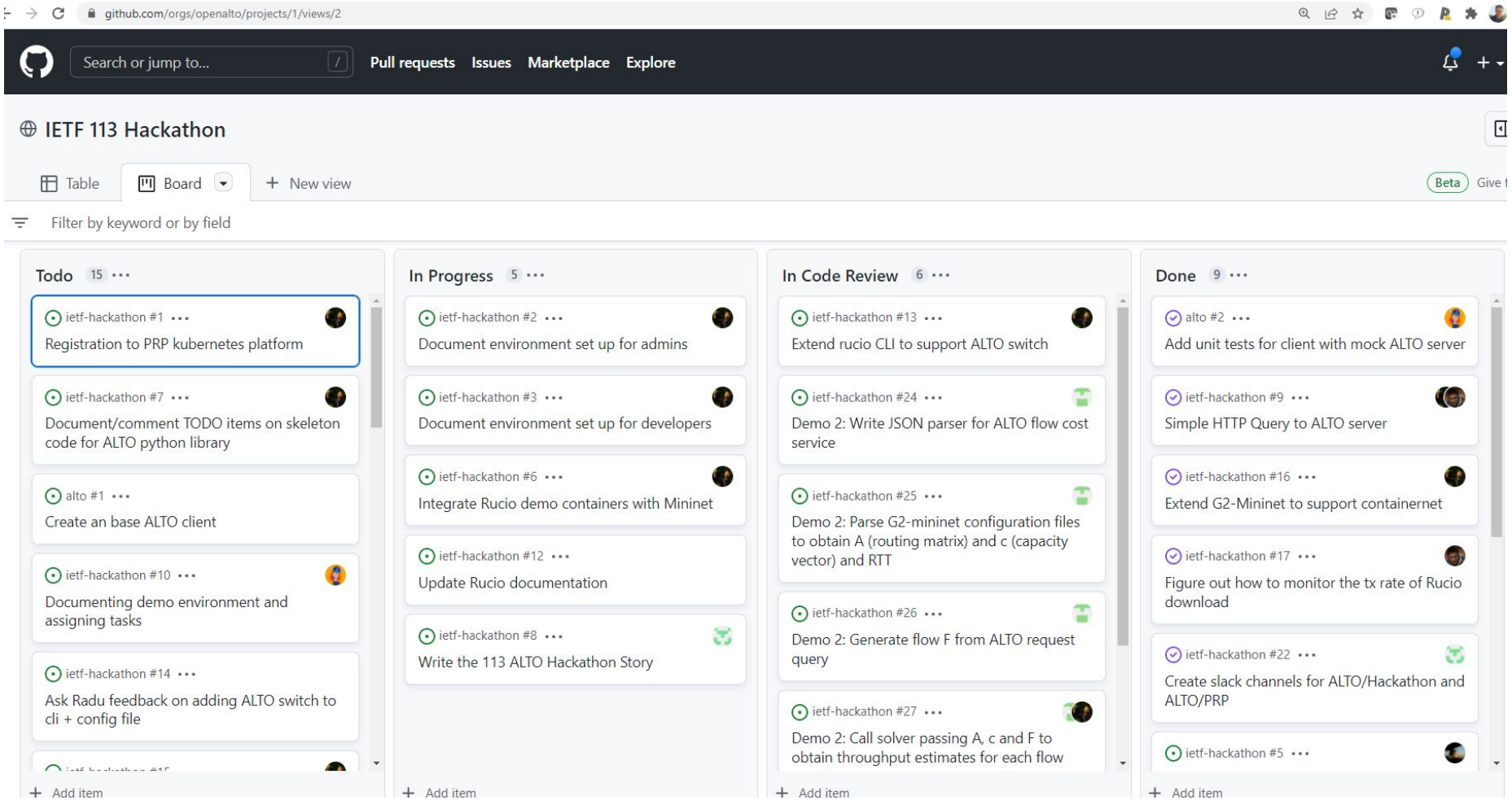


Project Management and Approach

- Adopting industry standard Agile / Scrum methodologies to ensure:
  - Productivity
  - Quality
  - Participation
  - Lean startup, rough consensus running code
- Invoke community participation to develop the ALTO Code Base:
  - Two roles: developers and mentors.
  - Mentors are usually experienced members of the IETF ALTO WG
  - Developers usually come from universities and the industry in general
- Project management resources:
  - Repo: https://github.com/openalto/
  - Project Scrum Dashboard (IETF Hackathon 113): https://github.com/orgs/openalto/projects/1

Project Management and Approach

Scrum dashboard:



ALTO Deployments

ALTO Deployments

- Current implementations/deployments:
  - Wiki list of implementations: https://trac.ietf.org/trac/alto/wiki/Impl
  - Examples:
    - Comcast P4P: RFC 5632: https://datatracker.ietf.org/doc/html/rfc5632
    - Benocs: https://people.csail.mit.edu/gsmaragd/publications/CoNEXT2019/CoNEXT2019.pdf
    - Telefonica: https://dl.ifip.org/db/conf/im/im2021mini/212012.pdf
- Forthcoming new deployments (work in progress):
  - Pacific Research Platform
  - CERN Rucio
  - UCSD 5G
  - NY City Cosmos 5G
  - ESnet

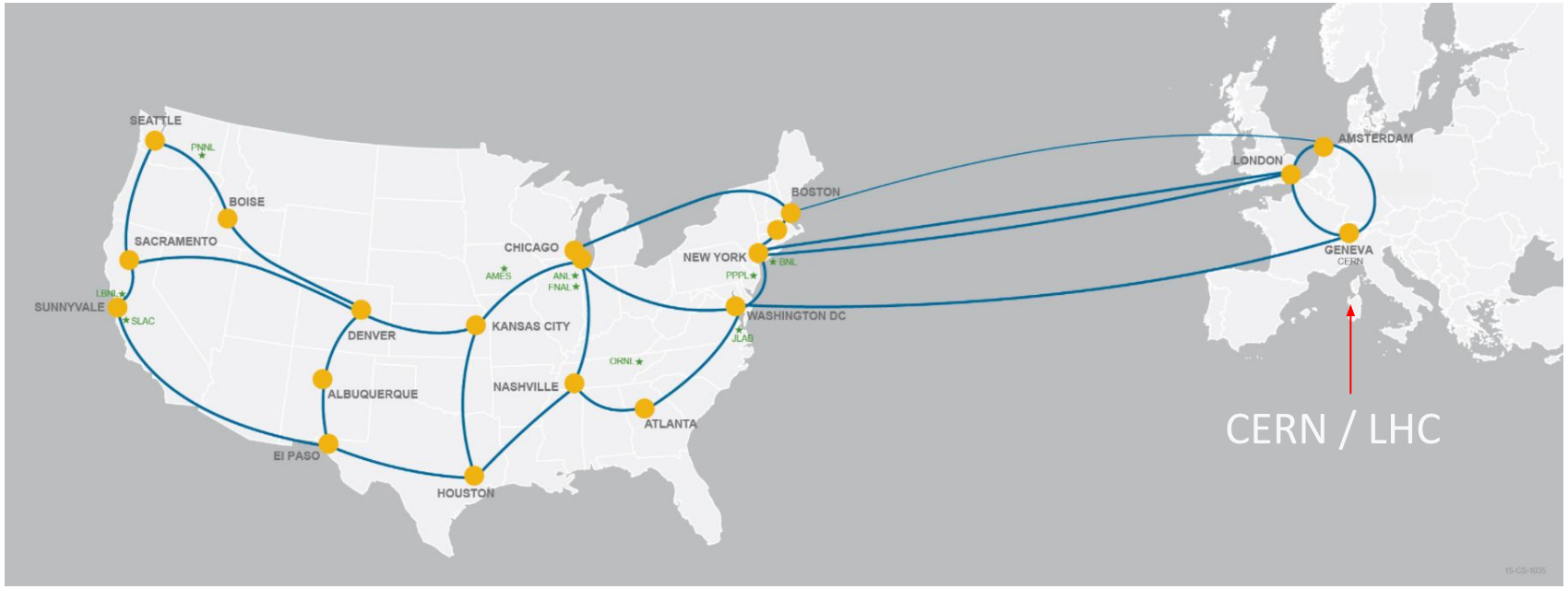
IETF Hackathon 113: ALTO Project

Summary

- Demo setup
  - Network is simulated by Mininet
  - Applications are running as virtual hosts/containers
  - The demo environment is packed as multiple containers for future enhancement
- Capabilities
  - Source selection based on network map and cost map
  - Costs computed as inverse of bandwidth
- Code base
  - An open-source python client library is developed
  - Add the ALTO-based replica selection support to the Rucio scientific data management system

Goals in the 113 Hackathon

- Use ALTO Cost Maps to optimize dataset transfers for rucio, the main data management tool for LHC and other large projects.
- Integrate ALTO Northbound Interface with Rucio to provide visibility and achieve better performance.
- Show that it works.



* ESnet / LHCONE source: https://www.es.net/about/

RFCs Involved During the Hackathon

- RFC 7285: Application-Layer Traffic Optimization (ALTO) Protocol
  https://datatracker.ietf.org/doc/rfc7285/
- I-Draft ALTO Extension: Flow-based Cost Query
  https://datatracker.ietf.org/doc/draft-gao-alto-fcs/
- I-Draft ALTO Performance Cost Metrics
  https://datatracker.ietf.org/doc/draft-ietf-alto-performance-metrics/

What Got Done

- Implementation of an ALTO Client in Python (RFC 7285)
- Integration with CERN Rucio replica download
  - Submitted pull request to Rucio Project:
    - https://github.com/rucio/rucio/pull/5364
- 3 Demos [https://github.com/openalto/ietf-hackathon/issues/8]
  - [D1] Single-flow replica node selection using ALTO BW Cost Map
  - [D2] ALTO Estimator: Multi-flow BW prediction
  - [D3] ALTO Scheduler: SLA-constrained multi-flow node selection
- Southbound ALTO integration with SDN:
  - Mininet/Pox, OpenDaylight
- Scrum dashboard: https://github.com/orgs/openalto/projects/1/views/1
- Lots of really interesting architecture discussions

ALTO Metrics

https://datatracker.ietf.org/doc/draft-ietf-alto-performance-metrics/

Metric	Definition in this doc	Semantics Based On
One-way Delay	Section 3.1	Base: [RFC7471,8570,8571] sum Unidirectional Delay
Round-trip Delay	Section 3.2	Base: Sum of two directions from above
Delay Variation	Section 3.3	Base: [RFC7471,8570,8571] sum of Unidirectional Delay Variation
Loss Rate	Section 3.4	Base: [RFC7471,8570,8571] sum Unidirectional Link Loss
Residual Bandwidth	Section 4.2	Base: [RFC7471,8570,8571] min Unidirectional Residual BW
Available Bandwidth	Section 4.3	Base: [RFC7471,8570,8571] min Unidirectional Avail. BW
Utilized Bandwidth	Section 4.4	Base: [RFC7471,8570,8571] max Unidirectional Utilized BW
TCP Throughput	Section 4.1	[RFC8312bis]
Hop Count	Section 3.5	[RFC7285]

Table 1. Cost Metrics Defined in this Document.

Metrics used in this hackathon

IETF ALTO Code Base

Container Environment Description

`docker-compose-with-rucio.yml`

It provides the following containers:

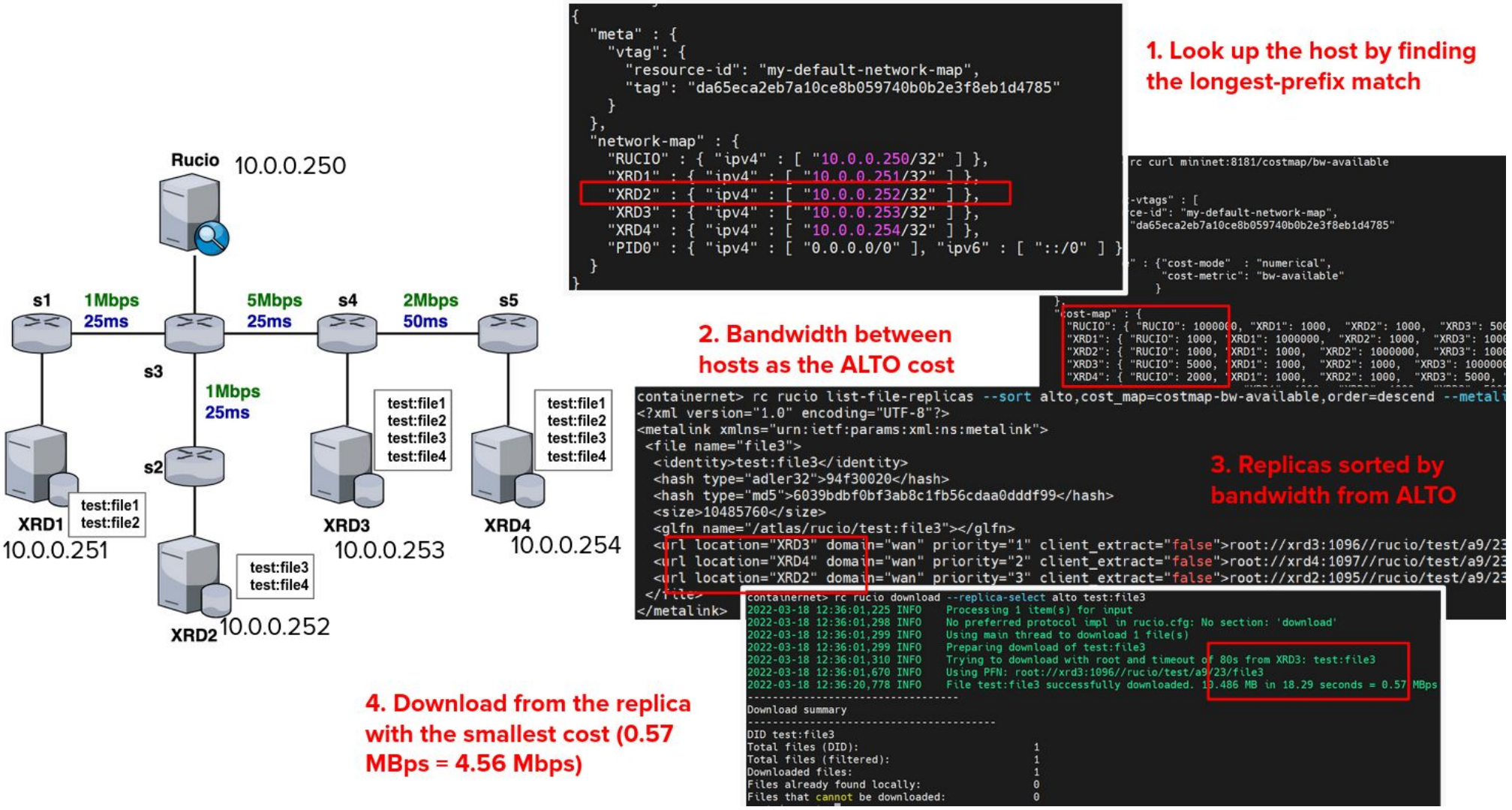
Service/Container Name	Scope	Short Description
mininet	Mininet	Extended mininet container for network simulation
sflow	Mininet	Network flow sampling and monitoring
odl	ALTO	ALTO server over OpenDaylight controller
rucio	Rucio	Basic Rucio components
ruciodb	Rucio	ProgresSQL database for Rucio
fts	Rucio	File transfer system for Rucio
ftsdb	Rucio	MySQL database for fts
activemq	Rucio	Message queue for Rucio and fts scheduler
xrd{i}	Storage	Xrootd storage node as Rucio RSE
ssh{i}	Storage	SSH storage node as Rucio RSE
minio	Storage	MinIO storage node as Rucio RSE
graphite	Monitoring	Graphtie server for Rucio internal monitoring

`docker-compose-with-rucio-monit.yml`

Besides the containers above, it also provides the following additional containers:

Service/Container Name	Scope	Short Description
elasticsearch	Monitoring	ElasticSearch engine for more complicated monitoring data processing
kibana	Monitoring	Kibana dashboard for customized visualization
logstash	Monitoring	Data processing pipeline for ElasticSearch
grafana	Monitoring	Another dashboard for visualization

Demo 1: Single-flow Replica Node Selection Using ALTO BW Cost Map



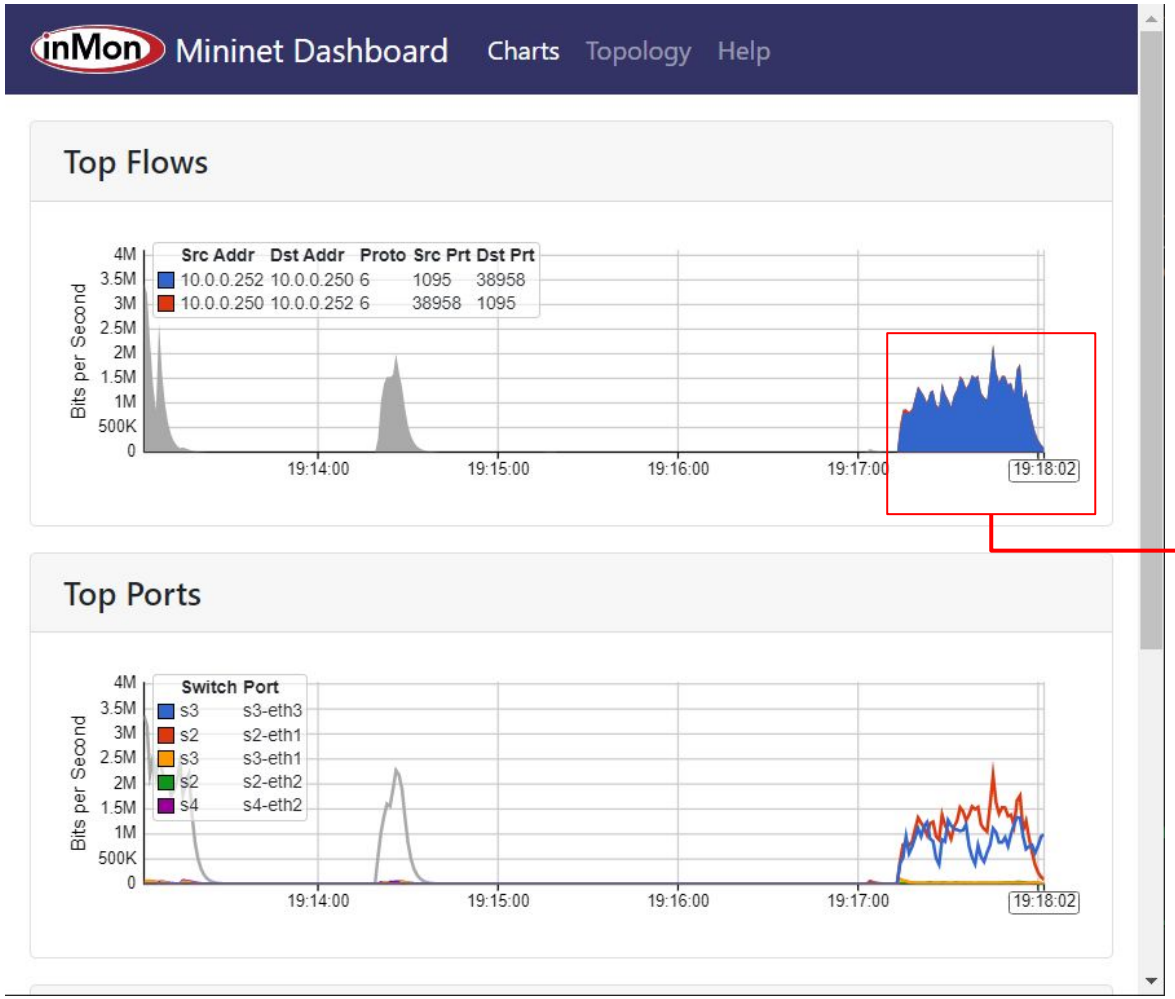

```
{
  "meta" : {
    "vtag": {
      "resource-id": "my-default-network-map",
      "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
    }
  },
  "network-map" : {
    "RUCIO" : { "ipv4" : [ "10.0.0.250/32" ] },
    "XRD1" : { "ipv4" : [ "10.0.0.251/32" ] },
    "XRD2" : { "ipv4" : [ "10.0.0.252/32" ] },
    "XRD3" : { "ipv4" : [ "10.0.0.253/32" ] },
    "XRD4" : { "ipv4" : [ "10.0.0.254/32" ] },
    "PID0" : { "ipv4" : [ "0.0.0.0/0" ], "ipv6" : [ "::/0" ] }
  }
}
```

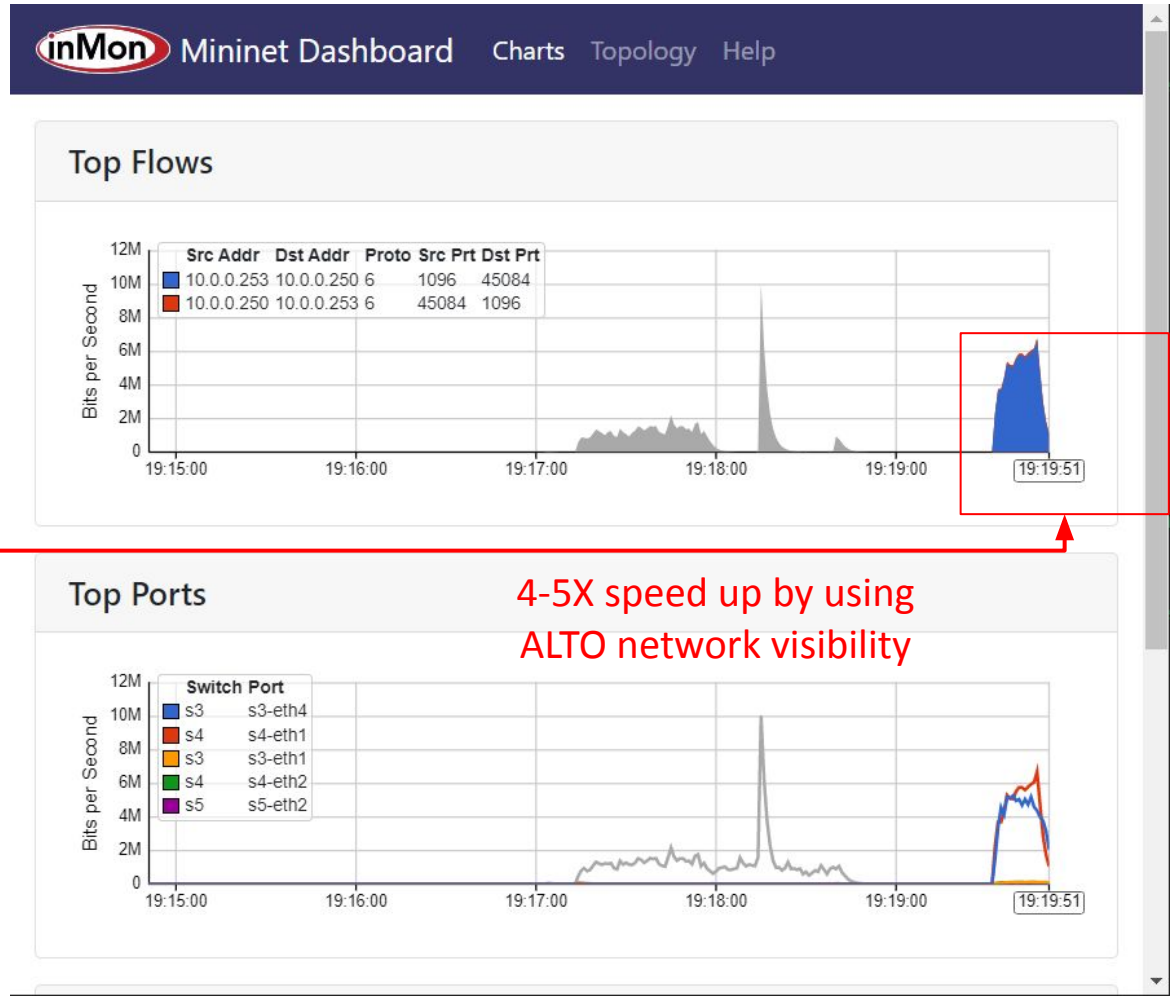
```
rc curl mininet:8181/costmap/bw-available

-vtags" : [
ce-id": "my-default-network-map",
"da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"

" : {"cost-mode"  : "numerical",
    "cost-metric": "bw-available"
  }
}
"cost-map" : {
 "RUCIO": { "RUCIO": 1000000, "XRD1": 1000,  "XRD2": 1000,  "XRD3": 500
 "XRD1": { "RUCIO": 1000, "XRD1": 1000000,  "XRD2": 1000,  "XRD3": 100
 "XRD2": { "RUCIO": 1000, "XRD1": 1000,  "XRD2": 1000000,  "XRD3": 100
 "XRD3": { "RUCIO": 5000, "XRD1": 1000,  "XRD2": 1000,  "XRD3": 1000000
 "XRD4": { "RUCIO": 2000, "XRD1": 1000,  "XRD2": 1000,  "XRD3": 5000,
```

```
containernet> rc rucio list-file-replicas --sort alto,cost_map=costmap-bw-available,order=descend --metali
<?xml version="1.0" encoding="UTF-8"?>
<metalink xmlns="urn:ietf:params:xml:ns:metalink">
 <file name="file3">
  <identity>test:file3</identity>
  <hash type="adler32">94f30020</hash>
  <hash type="md5">6039bdbf0bf3ab8c1fb56cdaa0dddf99</hash>
  <size>10485760</size>
  <glfn name="/atlas/rucio/test:file3"></glfn>
  <url location="XRD3" domain="wan" priority="1" client_extract="false">root://xrd3:1096//rucio/test/a9/23
  <url location="XRD4" domain="wan" priority="2" client_extract="false">root://xrd4:1097//rucio/test/a9/23
  <url location="XRD2" domain="wan" priority="3" client_extract="false">root://xrd2:1095//rucio/test/a9/23
 </file>
</metalink>
```

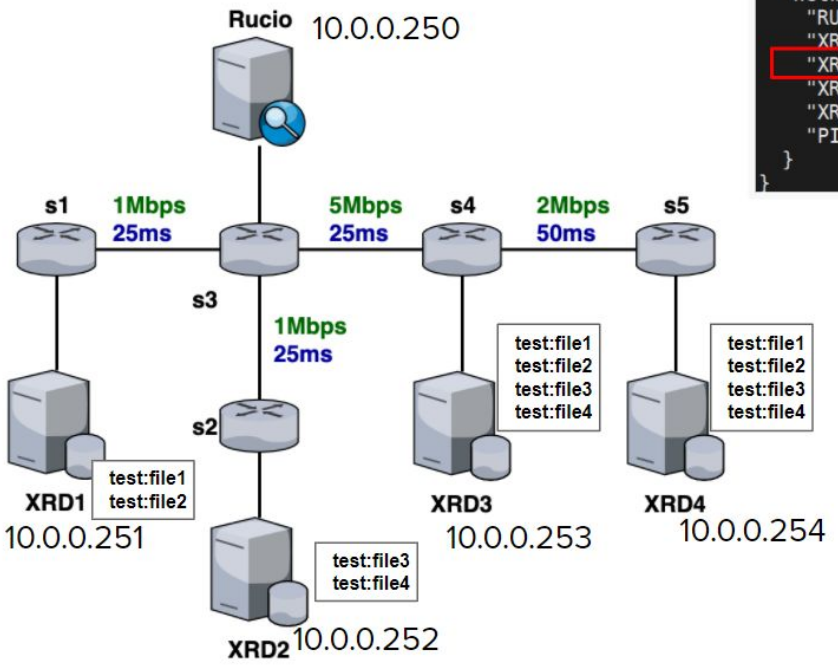
```
containernet> rc rucio download --replica-select alto test:file3
2022-03-18 12:36:01,225 INFO    Processing 1 item(s) for input
2022-03-18 12:36:01,298 INFO    No preferred protocol impl in rucio.cfg: No section: 'download'
2022-03-18 12:36:01,299 INFO    Using main thread to download 1 file(s)
2022-03-18 12:36:01,299 INFO    Preparing download of test:file3
2022-03-18 12:36:01,310 INFO    Trying to download with root and timeout of 80s from XRD3: test:file3
2022-03-18 12:36:01,670 INFO    Using PFN: root://xrd3:1096//rucio/test/a9/23/file3
2022-03-18 12:36:20,778 INFO    File test:file3 successfully downloaded. 10.486 MB in 18.29 seconds = 0.57 MBps
---------------------------------
Download summary
-----------------------------------------
DID test:file3
Total files (DID):                      1
Total files (filtered):                 1
Downloaded files:                       1
Files already found locally:            0
Files that cannot be downloaded:        0
```

Demo 1: Single-flow Replica Node Selection Using ALTO BW Cost Map





Demo 1: Single-flow Replica Node Selection Using ALTO Latency Cost Map




```
{
  "meta" : {
    "vtag": {
      "resource-id": "my-default-network-map",
      "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
    }
  },
  "network-map" : {
    "RUCIO" : { "ipv4" : [ "10.0.0.250/32" ] },
    "XRD1" : { "ipv4" : [ "10.0.0.251/32" ] },
    "XRD2" : { "ipv4" : [ "10.0.0.252/32" ] },
    "XRD3" : { "ipv4" : [ "10.0.0.253/32" ] },
    "XRD4" : { "ipv4" : [ "10.0.0.254/32" ] },
    "PID0" : { "ipv4" : [ "0.0.0.0/0" ], "ipv6" : [ "::/0" ] }
  }
}
```

1. Look up the host by finding the longest-prefix match

```
rc curl mininet:8181/costmap/delay-ow


-vtags" : [
ce-id": "my-default-network-map",
"da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"


" : {"cost-mode"  : "numerical",
    "cost-metric": "delay-ow"
    }

{
RUCIO": { "RUCIO": 0,  "XRD1": 25,  "XRD2": 25,  "XRD3": 25, "XRD4": 75, "PID0": 0
XRD1": { "RUCIO": 25,  "XRD1": 0,  "XRD2": 50,  "XRD3": 50, "XRD4": 100, "PID0": 2
XRD2": { "RUCIO": 25,  "XRD1": 50,  "XRD2": 0,  "XRD3": 50, "XRD4": 100, "PID0": 2
XRD3": { "RUCIO": 25,  "XRD1": 50,  "XRD2": 50,  "XRD3": 0, "XRD4": 50, "PID0": 25
XRD4": { "RUCIO": 75,  "XRD1": 100,  "XRD2": 100,  "XRD3": 50, "XRD4": 0, "PID0":
PID0": { "RUCIO": 0,  "XRD1": 25,  "XRD2": 25,  "XRD3": 25, "XRD4": 75, "PID0": 0
```

2. One-way latency between hosts as the ALTO cost

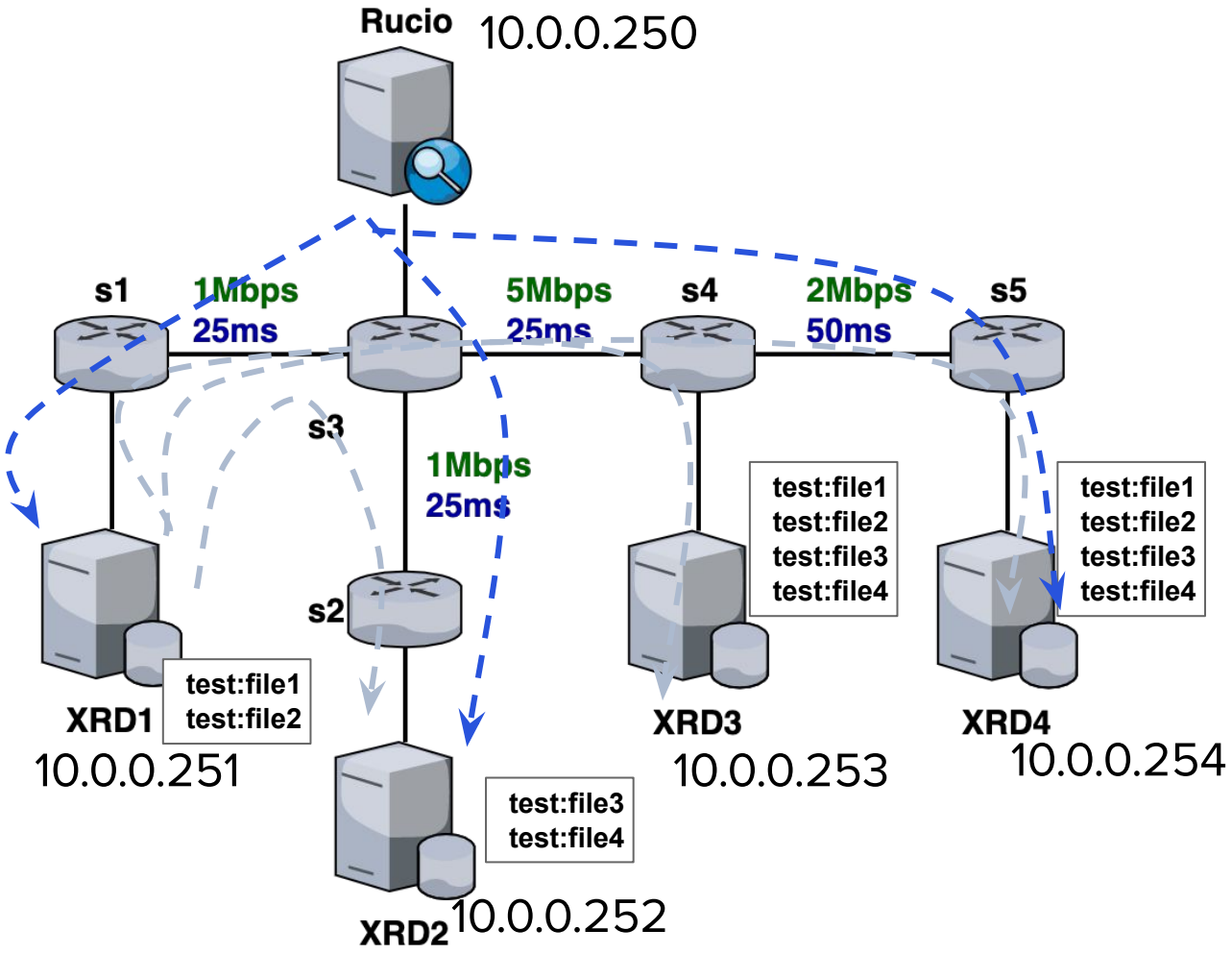
```
containernet> rc rucio list-file-replicas --sort alto,cost_map=costmap-delay-ow --metalink test:file3
<?xml version="1.0" encoding="UTF-8"?>
<metalink xmlns="urn:ietf:params:xml:ns:metalink">
 <file name="file3">
  <identity>test:file3</identity>
  <hash type="adler32">94f30020</hash>
  <hash type="md5">6039bdbf0bf3ab8c1fb56cdaa0dddf99</hash>
  <size>10485760</size>
  <glfn name="/atlas/rucio/test:file3"></glfn>
  <url location="XRD2" domain="wan" priority="1" client_extract="false">root://xrd2:1095//rucio/test/
  <url location="XRD3" domain="wan" priority="2" client_extract="false">root://xrd3:1096//rucio/test/
  <url location="XRD4" domain="wan" priority="3" client_extract="false">root://xrd4:1097//rucio/test/
 </file>
</metalink>
```

3. Replicas sorted by latency from ALTO

```
2022-03-20 11:16:51,737 INFO    No preferred protocol impl in rucio.cfg: No section: 'download'
2022-03-20 11:16:51,738 INFO    Using main thread to download 1 file(s)
2022-03-20 11:16:51,738 INFO    Preparing download of test:file3
2022-03-20 11:16:51,755 INFO    Trying to download with root and timeout of 80s from XRD2: test:file3
2022-03-20 11:17:02,569 INFO    Using PFN: root://xrd2:1095//rucio/test/a9/23/file3
2022-03-20 11:18:42,144 INFO    File test:file3 successfully downloaded. 10.486 MB in 98.75 seconds = 0.11 MBps
---------------------------------
Download summary
---------------------------------------
DID test:file3
Total files (DID):                      1
Total files (filtered):                 1
```

4. Download from the replica with the smallest latency

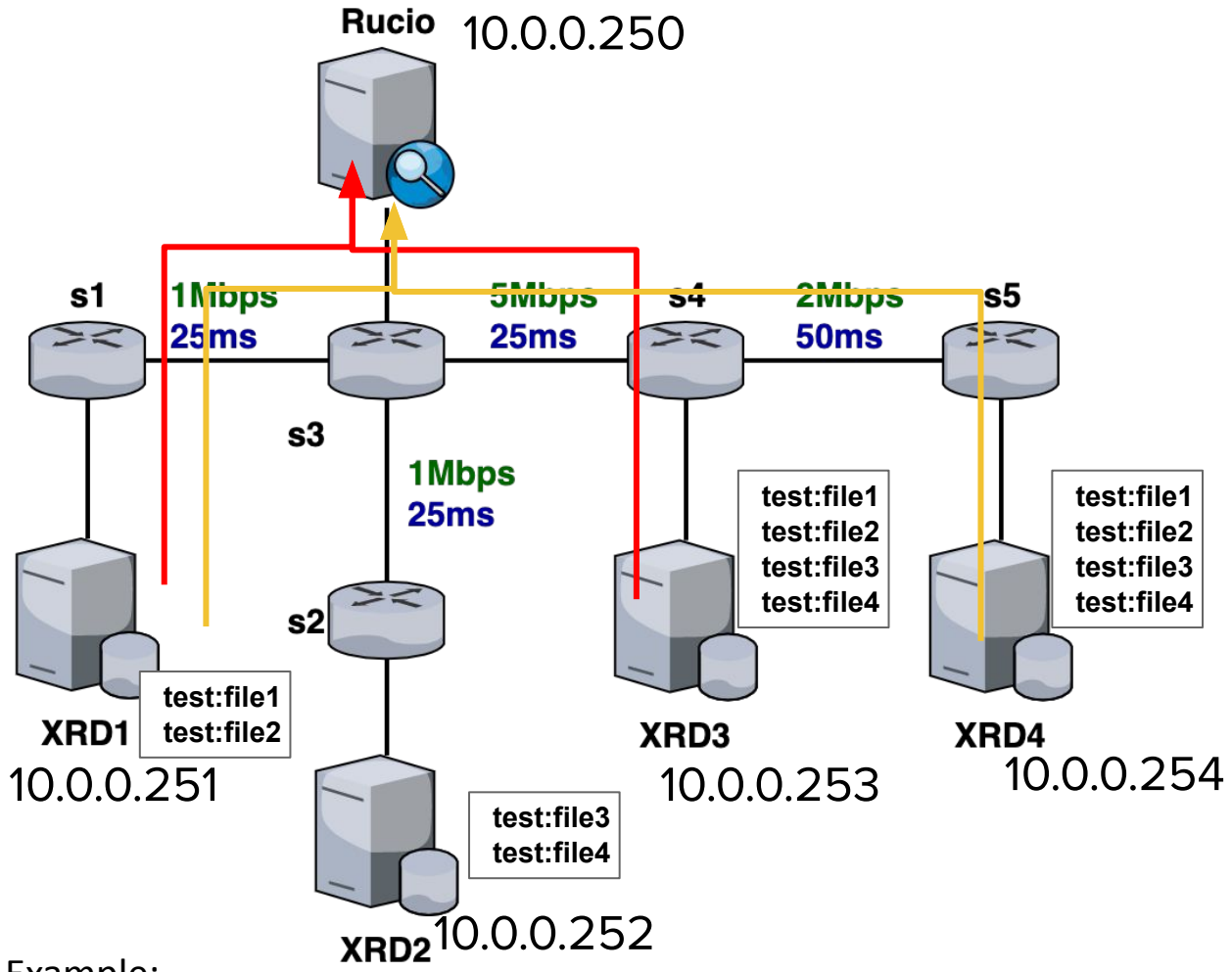
Demo 2: ALTO Estimator: Multi-flow Tput/BW Prediction




```
containernet> rc alto-estimator --alto-server http://minin
et:8181 --flows /opt/alto/tests/bwest/flows.demo | python3
 -m json.tool
{
    "meta": {
        "cost-type": {
            "cost-mode": "numerical",
            "cost-metric": "tput"
        }
    },
    "endpoint-cost-map": {
        "ipv4:10.0.0.250": {
            "ipv4:10.0.0.251": 0.2221213202250089,
            "ipv4:10.0.0.252": 0.4420177233358019,
            "ipv4:10.0.0.254": 0.335860959248392
        },
        "ipv4:10.0.0.251": {
            "ipv4:10.0.0.252": 0.2738144686147328,
            "ipv4:10.0.0.253": 0.2738145448368706,
            "ipv4:10.0.0.254": 0.23024966972090757
        }
    }
}
```

TCP throughput computed from network topology and TCP throughput modeling for bulk flows ([G2, PROPHET]).

Demo 3: ALTO Scheduler: SLA-constrained Multi-flow Node Selection



Example:
- Goal: download datasets file1, file3
- Replica selections: red versus yellow. Pick one replica that satisfies the SLA.

- Problem
  - Multiple datasets replicated on multiple hosts.
  - Rucio dataset automation workflow requires a given SLA (e.g., time-bound constrained data transfers)
- Demo **(partially finished)**
  - ALTO ESTIMATOR (Demo) provides cost map predicting replication throughput
  - ALTO SCHEDULER searches among possible download configuration one that guarantees the SLA requirement.

Wrap Up and Looking Forward

- **ALTO WG Contact:**
  - IETF ALTO WG: https://datatracker.ietf.org/wg/alto/about/
- **ALTO Code Base Project:**
  - Repo: https://github.com/openalto/
  - IETF Hackathon 113 ALTO Scrum Dashboard: https://github.com/orgs/openalto/projects/1/views/1
- **Potential Tasks/demos at IETF 114 hackathon:**
  - Finishing Demo 3, ALTO with HTTP/2
  - ALTO for multiple experiments for Rucio and more production use cases
- **Want to contribute to OpenALTO as a developer?** Reach us out: **jros at qti.qualcomm.com**

Looking forward to seeing you in Philadelphia!