Research Network Technical WG Update

scitags.org

Marian Babik (CERN), Shawn McKee (Univ. of Michigan)
net-wg@cern.ch | www.scitags.org
On behalf of the Research Networking Technical Working Group
HEPiX Autumn 2021 Workshop

Research Network Technical WG

The Research Networking Technical Working Group(RNTWG) was formed in response to the needs expressed by the WLCG experiments at the January 2020 LHCONE/LHCOPN meeting and has three main work areas:

1. Packet Marking
2. Traffic Shaping
3. Network Orchestration

WG has now 97 members from ~ 50 organisations

Packet Marking Subgroup

Packet marking was viewed as the appropriate first step and we have had regular meetings every ~2 months since summer 2020.

It is strongly motivated by the need to understand the source of network flows seen on our R&E networks.
We have seen cases of new traffic increasing link usage by 100% for 10's of hours but we were not easily able understand the source. Is this an unintentional side product of an experiment tweaking one of their work-flow configs or distributed data management settings?

There are detailed results of our work in our Packet Marking Subgroup notes.

- **Goal** is to make flows identifiable anywhere along its path
- **Summary**: Packet marking will require IPv6 and newer kernels.
- It is possible to describe a flow via a side-channel: send special UDP packets ("Fireflies" suggested by ESnet) along the same path as the flow (works for IPv4 as well)

Motivation

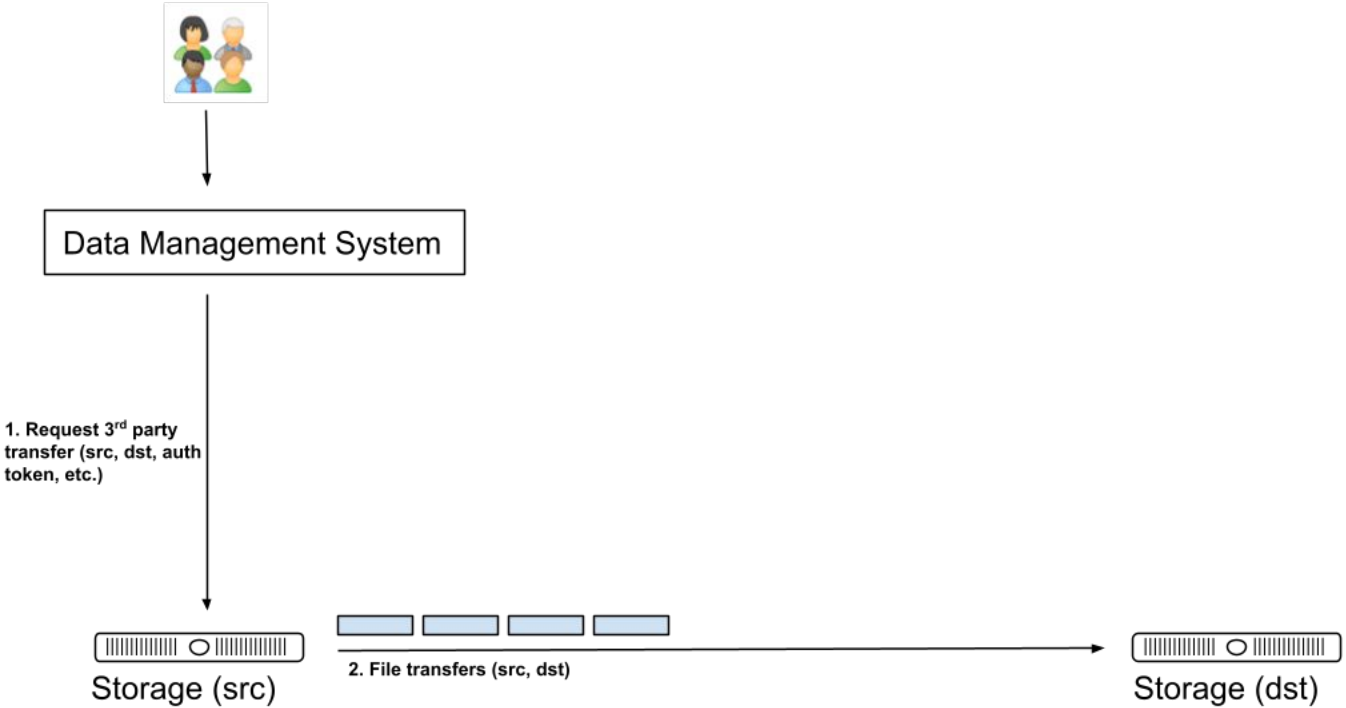
- Networks are becoming more programmable and capable with technologies such as P4, SDN, virtualisation, eBPF, etc.
- But with less and less context about the traffic they carry
  - Cloud deployments, Kubernetes, encryption, tunneling, privacy, etc.
- Understanding scientific traffic flows in detail is critical for understanding how our complex systems are actually using the network.
  - Current monitoring/logging tell us where data flows start and end, but is unable to understand the data in flight.
  - Dedicated L3VPNs can be created to track high throughput science domains, but with more domains requiring high throughput this will become expensive, it won't scale, won't work at big sites having to support multiple domains at the same time
- In general the monitoring we have is experiment specific and very difficult to correlate with what is happening in the network. We suggest this is a general problem for users of the Research and Education Networks (RENs)

Scientific Network Tags (scitags) is an initiative promoting identification of the science domains and their high-level activities at the network level.

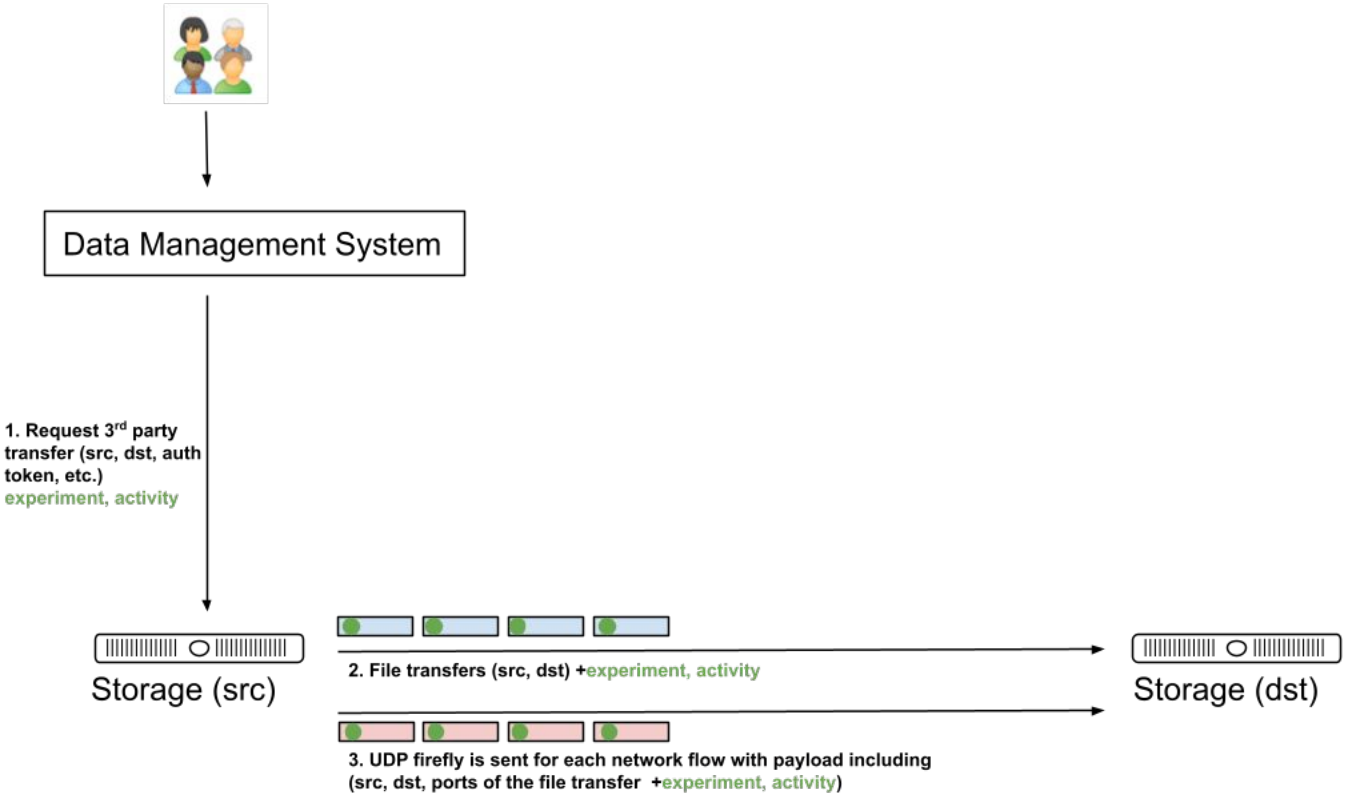
News

- New domain and web site (www.scitags.org)
- Flow and Packet Marking Technical Specification
- New github organisation (https://github.com/scitags)
  - Serves www.scitags.org via github pages
- Reference implementation
  - Flow service (flowd, https://github.com/scitags/flowd)
- Initial implementation in Xrootd
  - For both packet and flow marking
- Participation in the Data Challenge

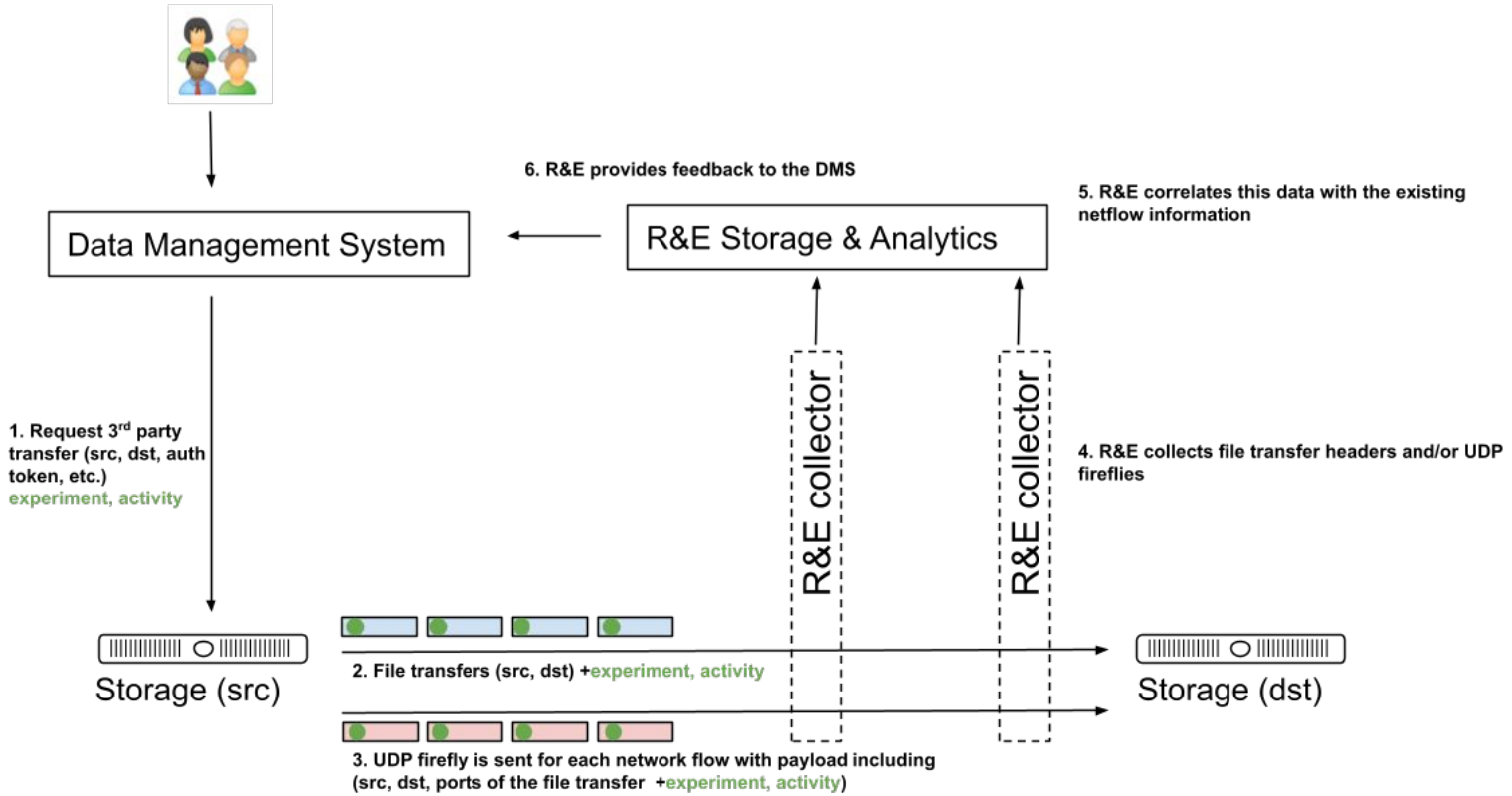
How scitags work



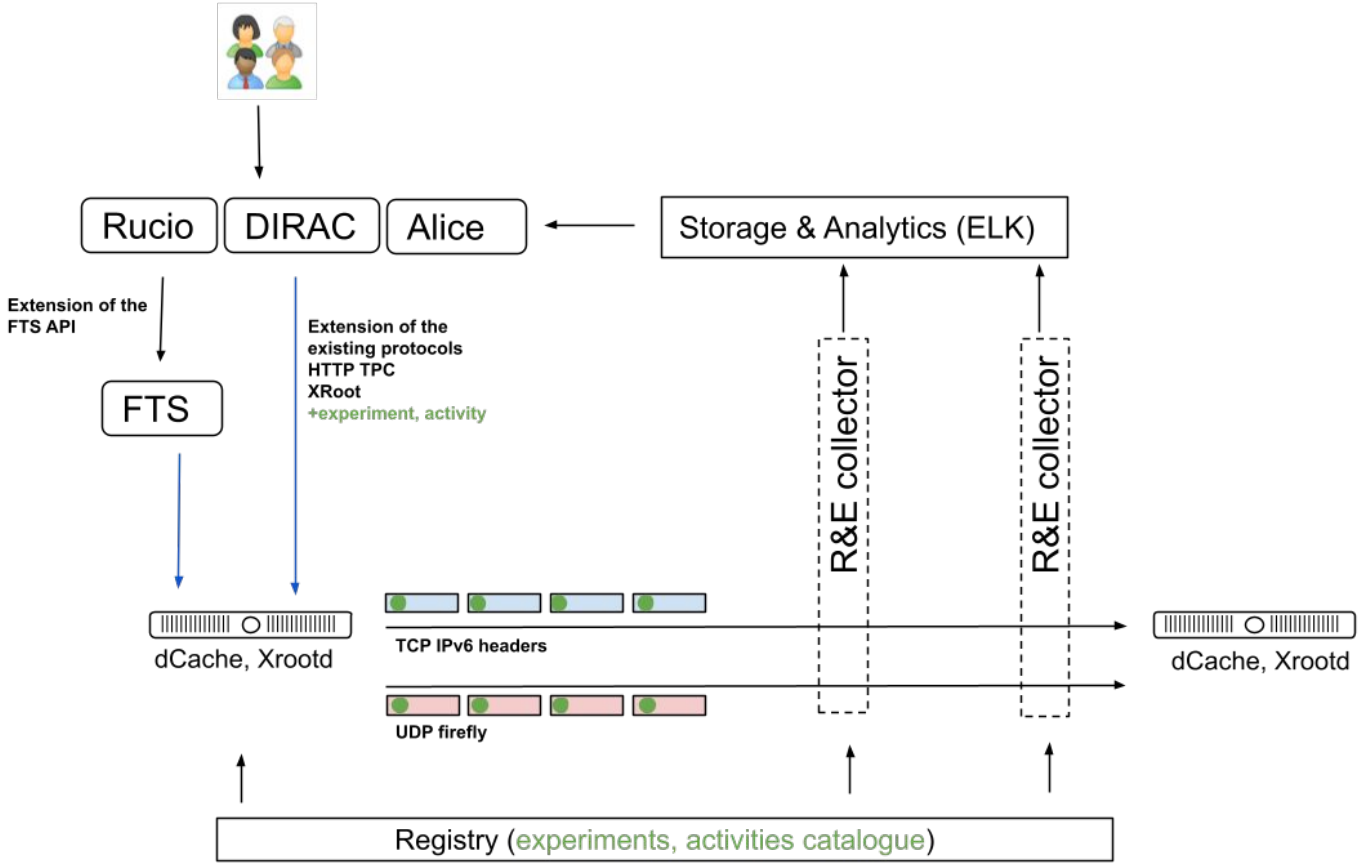
How scitags work



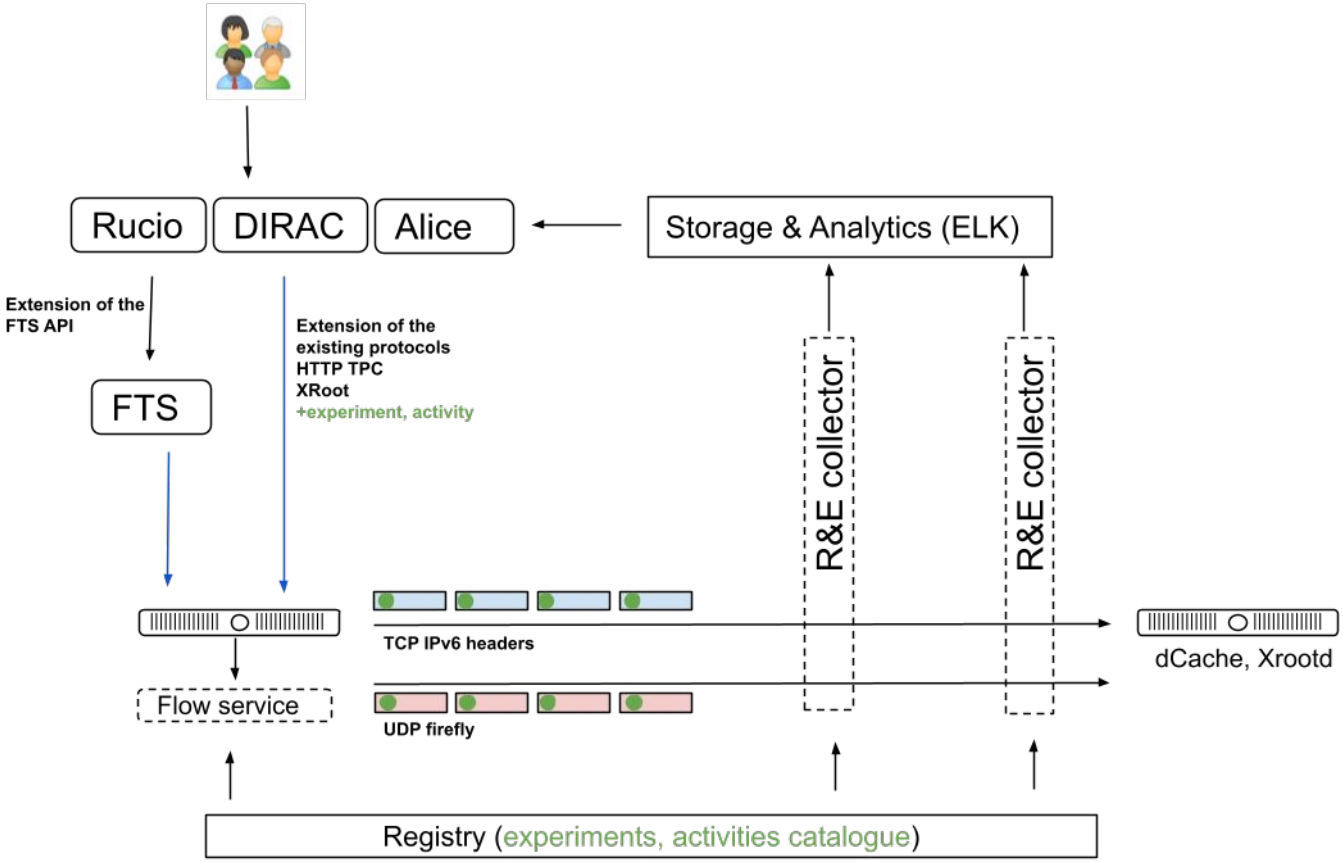
How scitags work



Scitags Architecture



Scitags Architecture



Concepts

- Marking is based on two different approaches
  - **Flow marking** using UDP fireflies (works for both IPv4 and IPv6)
  - **Packet marking** using IPv6 flow label and/or header extensions
- Both carry **flow identifier**, which at present is an encoded representation of experiment/science domain and activity
  - For UDP fireflies flow id can be extended with other fields in the future
  - For packet marking the space is restricted due to number of bits available in the headers
- Experiments and activities need to be registered prior to their usage
  - Registry serves this purpose and ensures RENs and DDMs have consistent view
- Designed to work with proxies, cached proxies and private networks
- Generators, collectors, storage and analytics can evolve independently

Technical Specification Updates

- UDP firefly packet specification
  - Dst port: 10514; Size: must fit into a single frame
  - Payload is a syslog message that conforms to RFC5424
    - Last part of the syslog message is a structured data specification (in JSON)
    - JSON schema for the structured data is also available in App. A
- Flow registry specification
  - Maps experiments and activities to IDs
  - Draft JSON schema, which is already used in the API
  - https://www.scitags.org/api.json
- Protocols updates
  - Xroot protocol extension with <scitag.flow> attribute to pass flow identifier as part of the URL
  - HTTP TPC protocol extension (passing flow identifier as part of the HTTP headers)

Reference Implementation

- **Flow service (flowd)** - developed to help test and validate the approach
  - Provides reference implementation of the technical specification
  - Storage systems can either provide their own implementation or use flowd
- Provides **pluggable** system to test different flow/packet marking strategies.
  - The following plugins currently exist:
    - **Netstat** - scans existing network connections (via netstat command) and assigns them a fixed experiment and activity id
    - **NP API** - provides local API (named pipe), which accepts connection information (src/dst, src/dst ports) along with the experiment and activity ids.
  - There is currently just one backend, which implements **UDP firefly** spec.
- Written in python; runs as a Linux service
  - Integrates with systemd/journal, supports CC7/C8/docker

- Possibility to add additional plugins
  - **Netlink**, which can provide additional information per connection (e.g. congestion algorithm, rtt/rto, cwnd, bytes sent/received/acked, rcv_rtt/min_rtt, delivery/pacing rates, etc.)
  - Hybrid plugins that combine information from storages with netstat/netlink also possible

WLCG Data Challenge

- Aim is to test and validate our approach in gradual steps, our initial goals:
  - Test flow service deployment directly on the site's storages
  - Generate UDP fireflies based on real traffic
  - Capture UDP packets (initially using a dedicated endpoint)
  - Understand how UDP firefly information can be correlated with R&E netflow data
- Flow service (flowd) deployment
  - **Currently deployed at AGLT2, BNL, KIT, UNL and Caltech**
  - Runs directly on the storage nodes, uses netstat plugin
  - Generates UDP firefly based on real traffic
- ESnet has setup a dedicated collector to capture the UDP fireflies
  - Will attempt to correlate them with their netflow data
- Results
  - Deployment, packet generation and collection worked fine
  - On-going - summary/results on the correlation with netflow

Plans

- Continue with the validation and testing using the existing deployment
  - Improve existing prototypes based on the feedback from the initial DC tests
  - Test and validate the possibility to run packet marking using flowd
- Extend testing to Xrootd once an official release is available
  - Find volunteer sites to test deploy directly on their storage nodes
  - Implementation supports both flow and packet marking
  - Can provide experiment IDs based on path access (supports multi-VO storages)
- Engage other R&Es and explore available technologies for collectors
  - Improve existing data collection and analytics
- Test and validate ways to propagate flow identifiers
  - Engage experiments and data management systems
  - Validate, test protocol extensions and FTS integration
  - Explore other possibilities for flow identifier propagation, e.g. tokens

Questions, comments ?

Draft Technical Specification available; Packet Marking Overview

Prototype testing as part of the WLCG Data Challenges effort in collaboration with ESnet

Prototype code of the flow service (flowd) implementing UDP fireflies

https://www.scitags.org