# Integrating IP Traffic Flow Measurement *

**Marcelo Pias & Jon Crowcroft**

Department of Computer Science
UCL - University College London
London, UK
{m.pias,j.crowcroft}@cs.ucl.ac.uk

**Juergen Quittek & Marcus Brunner**

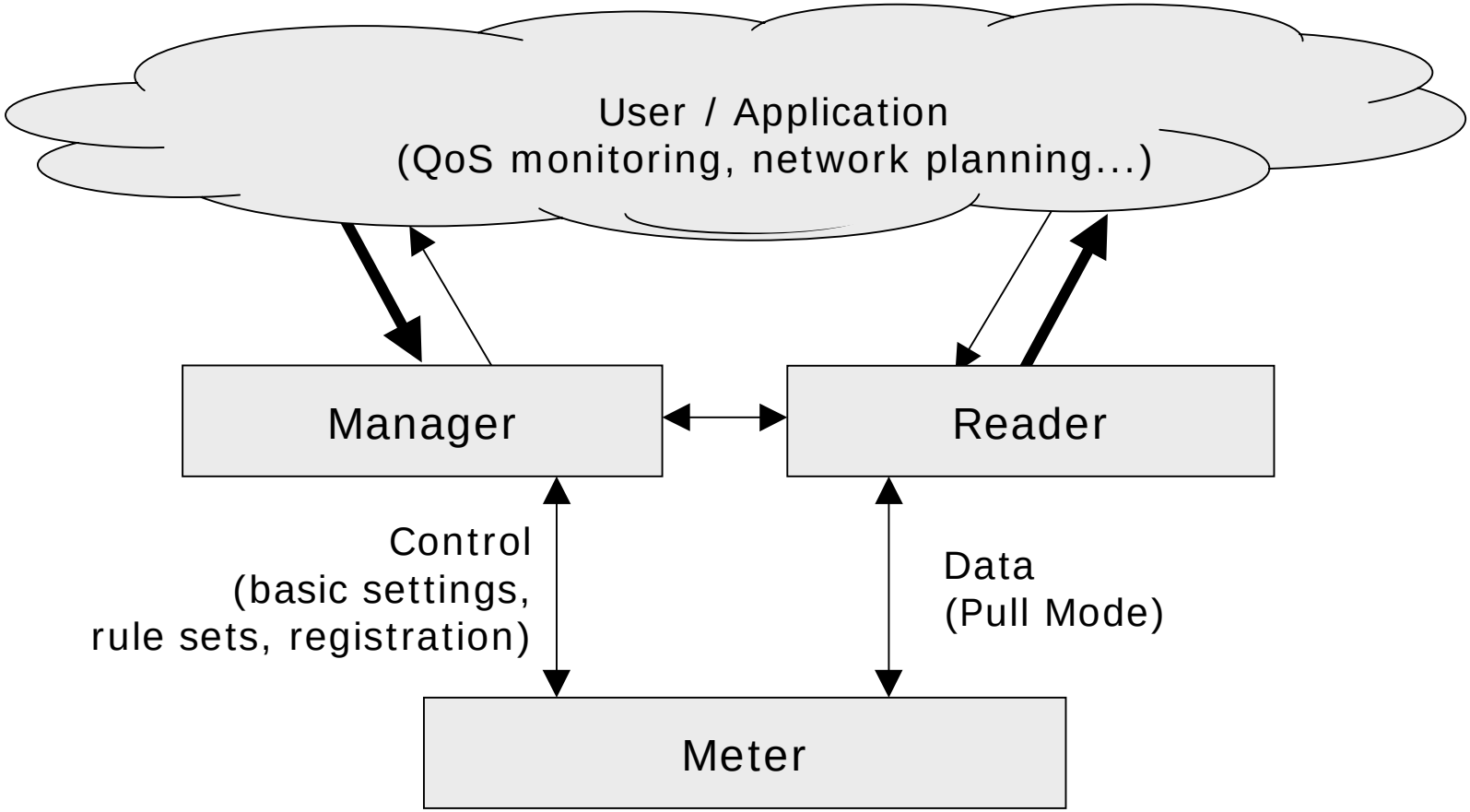
C&C Research Laboratories
NEC Europe Ltd.
Heidelberg, Germany
{quittek,brunner}@ccrle.nec.de

* Internet Draft presented at 49th IETF (San Diego) Dec 2000

## Realtime Flow Measurements architecture (RTFM)

- Defined by the IETF in RFC2722, Meter MIB [RFC2720]
- Objectives:
  - Specifies ways for flow measurements based on metrics useful for:
    - Understanding the behaviour of existing networks
    - Network planning for development and expansion
    - Quantification of network performance
    - Verifying the quality of network service (QoS)
    - Common ground for accounting in general
- Metrics:
  - Simple metrics: volume
  - Distributions: inter-arrival times, packet size, packet-rate...
  - List-valued: DSCodepoint...
  - Packet-header traces: ECN counters...
  - QoS metrics: delay, loss and jitter

# Introduction




- Architecture Implementation
  - NeTraMet (University of Auckland): free
  - IBM RTFM

# Motivation

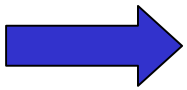
- The interface for flow measurements defined by the RTFM architecture and offered by the Meter MIB is powerful.

- However, it is not always really easy to have
  - interaction between manager, reader, and meter
  - rule set specifications are procedurally defined
  - writing rule sets is a non-trivial task
  - meter delivers measured traffic data in pull mode only

- Several management applications require less functionality, and they would benefit from a simpler interface.

# Rule Sets

- Single rule:
  <attribute> & <mask> = <value> : <action> , <index> ;
- Actions:
  Ignore, NoMatch, Count, CountPkt, Return, GoSub, GoSubAct, Assign, AssignAct, Goto, GotoAct, PushRuleTo, PushRuleToAct, PushPktTo, PushPktToAct
- Format statement
- Statistics statement
- Example



```
SET 13
RULES
  SourcePeerType & 255 = IP:        PushtoAct, IP_pkt;
  Null & 0 = 0:                      Ignore, 0;
#
IP_pkt:  # Tally IP traffic by (Class C) subnet
  SourcePeerAddress  & 255.255.255.0 = 0: PushPktToAct, Next;
  DestPeerAddress    & 255.255.255.0 = 0: CountPkt, 0;
#
FORMAT FlowRuleSet FlowIndex FirstTime "  "
   SourcePeerType SourcePeerAddress DestPeerAddress "  "
   SourceTransType SourceTransAddress DestTransAddress "  "
   ToPDUs FromPDUs "  " ToOctets FromOctets;
```

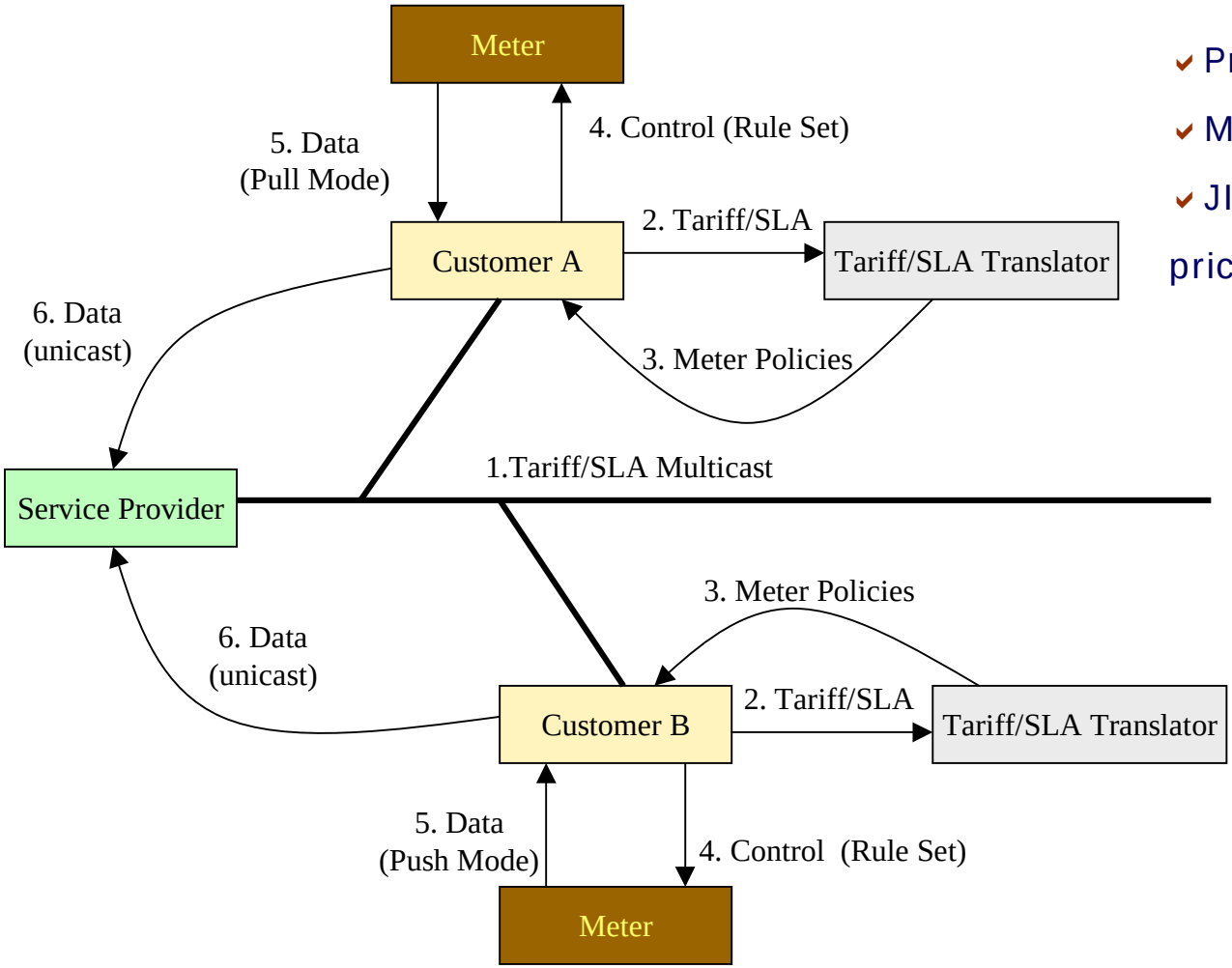
# RTFM Architecture Applications

**There are two typical kinds of applications:**

1 Plain (standalone) traffic measurement applications
- well supported by the architecture
- high flexibility in rule definition
- Rule files written manually or in SRL
- Traffic data is read in pull mode
- metered data stored for further processing

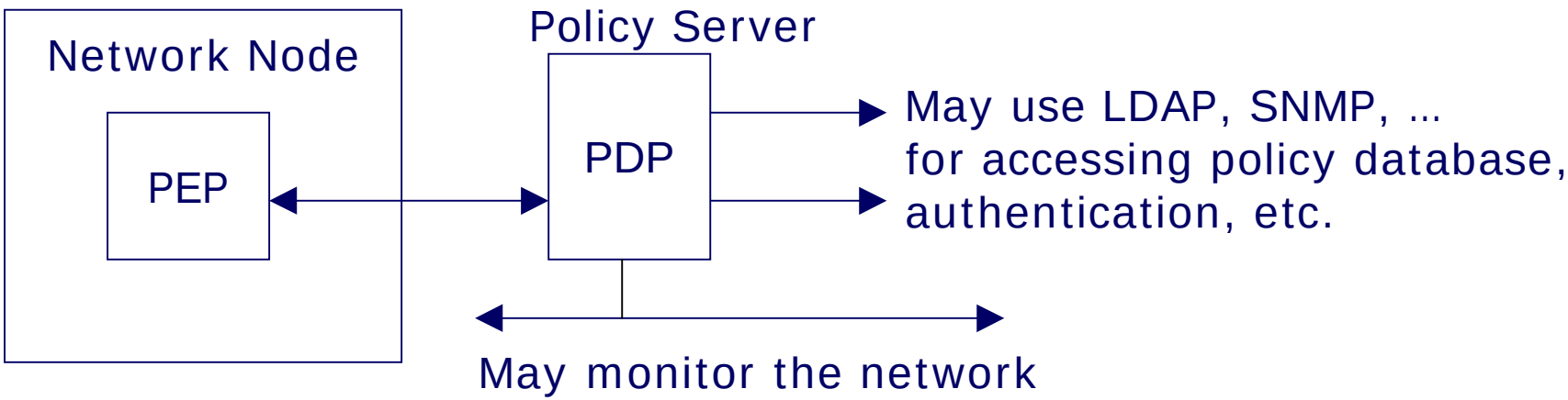
2 Applications with integrated traffic measurement
- typically less low-level metering functionality required
- automatic rule set generation
- push mode for traffic data may be desirable
- gathering of traffic data for immediate processing

# Accounting, Charging and QoS monitoring




- Projects
- M3I (HP, BT, TUD, ETHZ, Telenor)
- JISC Internet2:Congestion-based pricing (UCL, UCB)

# Policy Server

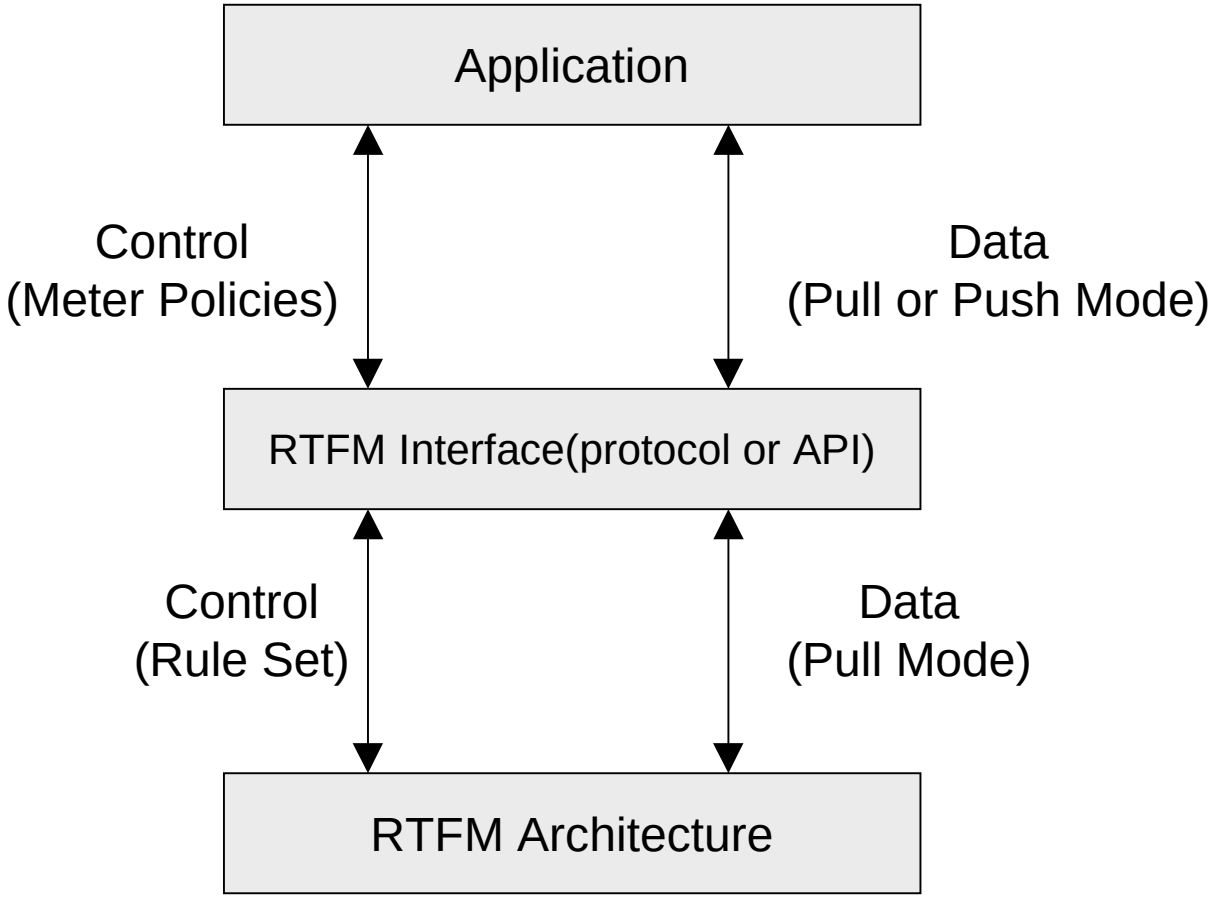



- Policy server according to policy framework RFC 2753
- May require current overall traffic and current individual traffic information in order to decide on admission control, QoS assignments, and traffic engineering actions.
- Projects:
  - European Union project MobiVAS where NEC is a partner

# Requirements for those Applications

- Control traffic measurement on a high level of abstraction
- Abstraction from details of RTFM architecture as high as possible
  - Simplification of model
  - Simplification of  meter policies (rule sets)
- Abstraction as low as necessary in order to provide the functionality required by different kinds of applications
  - Accounting
  - QoS monitoring
  - . . .
- Gathering traffic data in pull and push mode

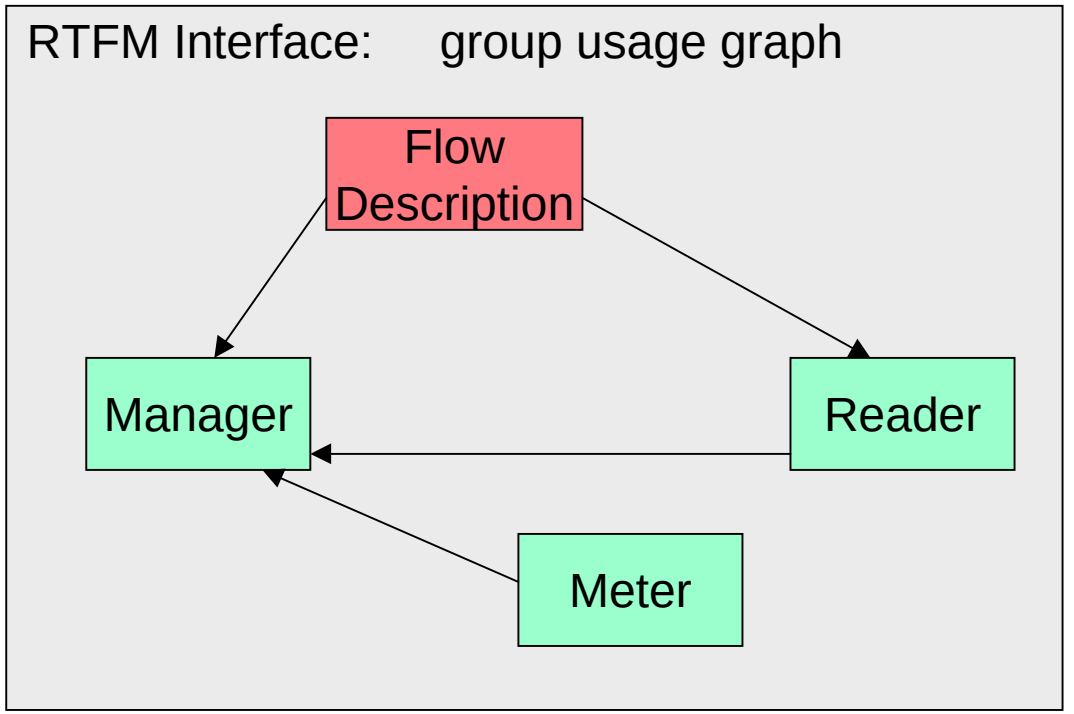
# Approach


Application
Control
(Meter Policies)
Data
(Pull or Push Mode)
RTFM Interface(protocol or API)
Control
(Rule Set)
Data
(Pull Mode)
RTFM Architecture

# Interface Design

- 5 Groups of data structures and functions
  - Flow Attribute
  - Flow Description
  - Manager
  - Reader
  - Meter

- Limitation:
  - Trade off in simplicity against functionality
  - Procedural x Declarative Flow Description



# Flow Description

- Declarative specification instead of procedural rule sets

| | | | |
|---|---|---|---|
| sourceIPAddress | destinationIPAddress | sourceMask | destinationMask |
| transportType | sourcePort | destinationPort | direction |
| FlowAttribute 1 | | | |
| FlowAttribute 2 | | | |
| ..... | | | |
| ..... | | | |
| FlowAttribute n | | | |

# Future Work

- **IETF work**
  - 49th IETF question: map to (Protocol, MIB, API or none)?
  - Decision on mapping to a protocol for exporting flow data from routers
  - IP Flow Export (IPFX) protocol
  - BOF for a new WG in the next IETF meeting (London Aug 2001)
  - IPFX WG contributions from:
    - NetFlow from Cisco
    - LFAP from Riverstone
    - RTFM architecture + this interface
- **Collaborative work with RNP2 WGs?**
  - Where this work might be helpful within the RNP2 WGs?
  - WGs currently working with QoS, traffic measurements and policy-based management?
  - The RTFM architecture impl. + this interface are freely available (GNU)
  - Contributions for the IPFX WG are welcome

# URLS

- M3I project
  - http://www.m3i.org
- RTFM Arch
  - http://www2.auckland.ac.nz/net/Internet/rtfm
- NeTraMet: RTFM implementation
  - http://www2.auckland.ac.nz/net/NeTraMet
- IETF I-D presented in these slides
  - http://www.cs.ucl.ac.uk/staff/rtfm
  - http://search.ietf.org/internet-drafts/draft-quittek-rtfm-generic-interface-00.txt
- Internet2/JISC Congestion-based Project
  - info available by Aug 2001