Scalable Software-Defined Monitoring

P. Sköldström[1], R. Steinert[2], C. Meirosu[3], W. John[3]

1 – ACREO Swedish ICT; 2 – SICS Swedish ICT; 3 – Ericsson Research

UNIFY is co-funded by the European Commission
DG CONNECT in FP7

Monitoring today

- Centralized and manual = scalability limitations





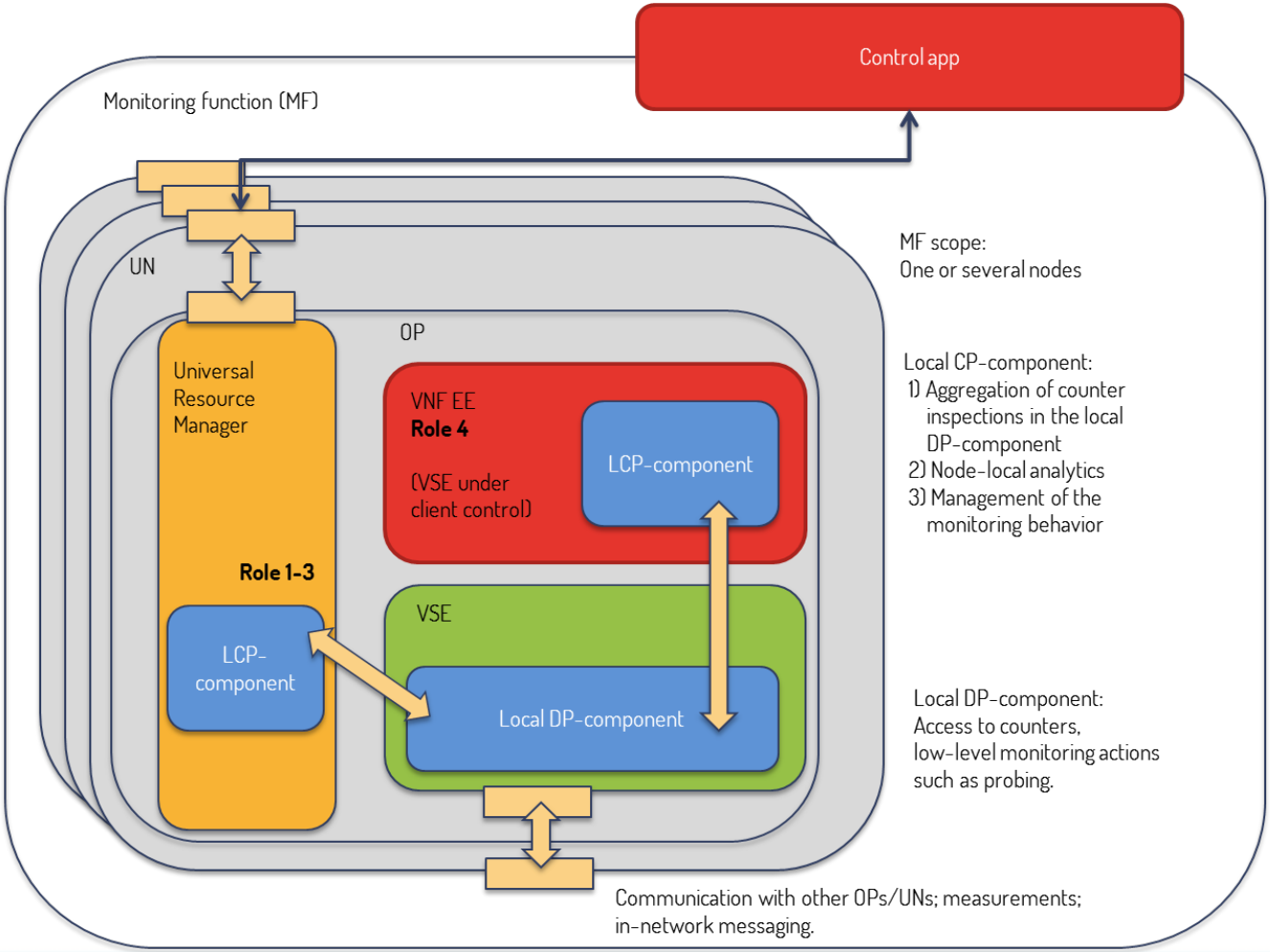
images source: Powerpoint ClipArt

The UNIFY project in a nutshell



- Increased velocity of service introduction
- Unified network-cloud programming abstraction: orchestration and generic processing
- Novel observability and verification features

Control-data split for monitoring functions



Programmability Advances Required

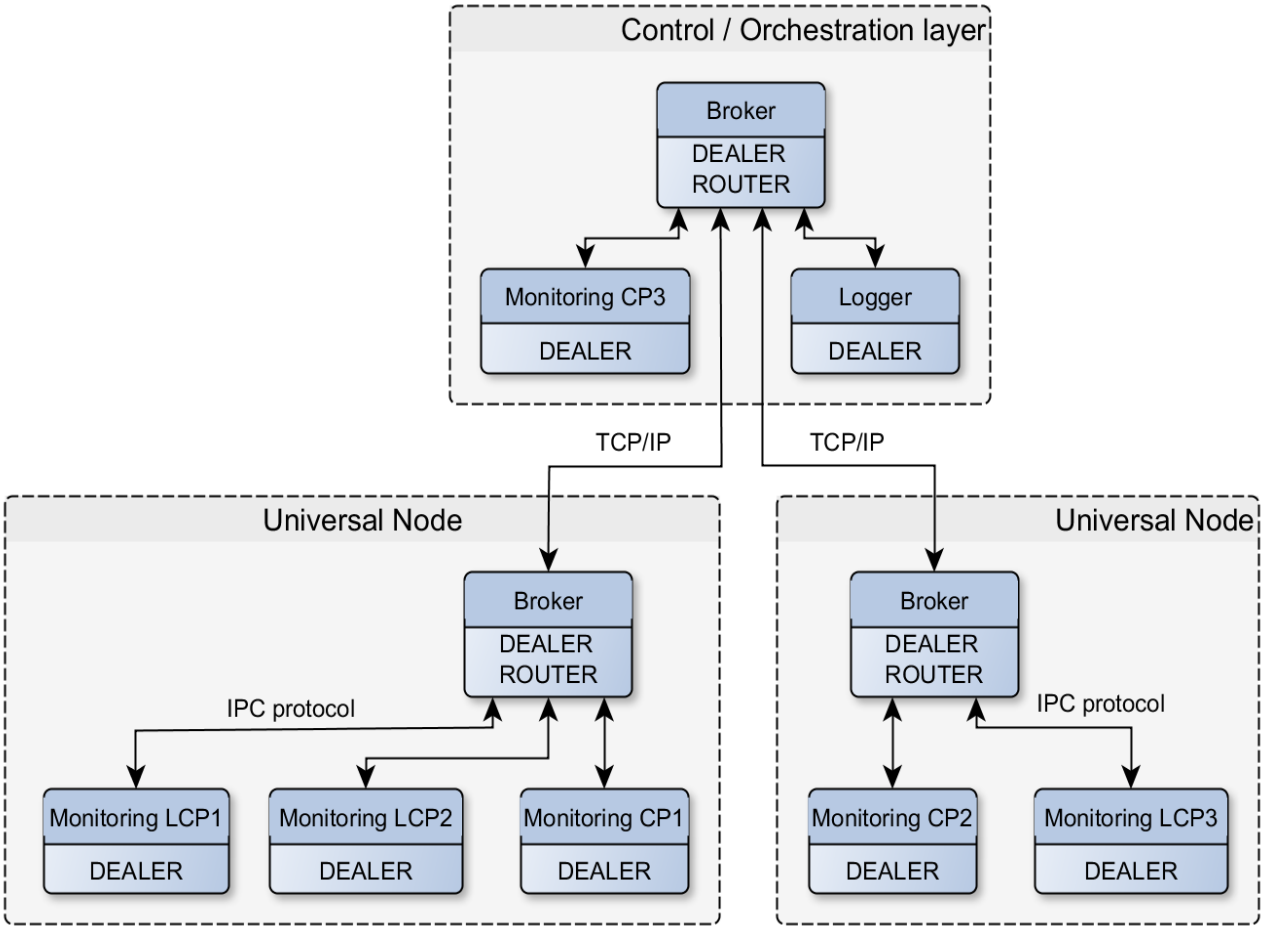
Major requirements for supporting the monitoring split architecture

- Access for deploying or implementing data plane monitoring components
- More expressive programming capabilities
- Better scalability for information dissemination
- Ease of integration with information consumers
  - APIs easily accessible to programmers and automated actors

ZeroMQ choice for a monitoring bus

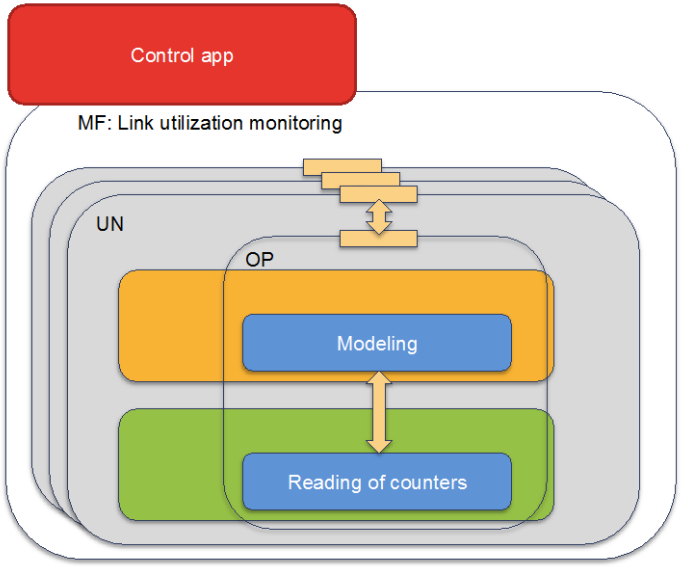
- Advantages
  - Multitude of transport protocols optimized for different communication scenarios:
    - inproc, IPC: same application, multiple threads
    - TCP/IP
    - TPIC
    - PGM reliable multicast
  - Wide support for programming language bindings
  - High throughput, low latency
- Disadvantages
  - No message persistency
  - No high-availability
  - No guaranteed ordering

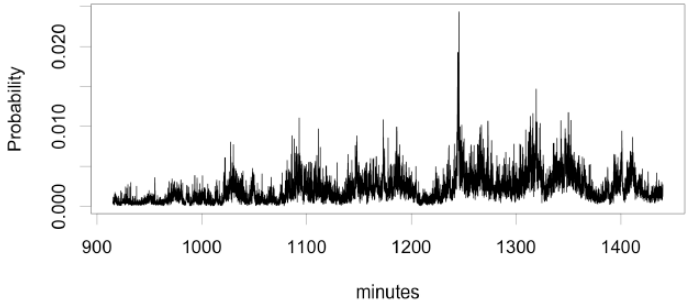
Example of implementing a ZeroMQ bus in UNIFY



Distributed link utilization monitor

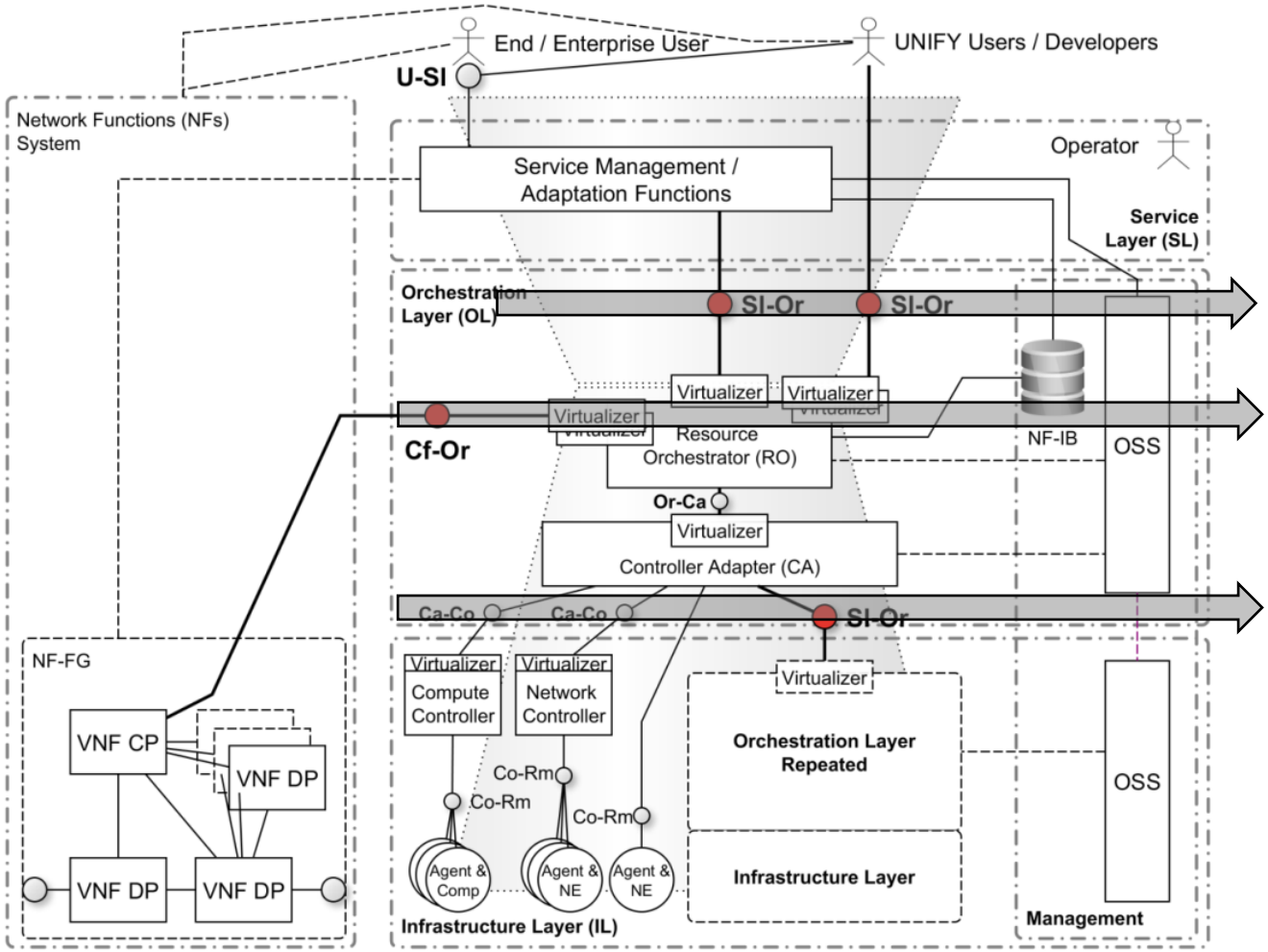
- Use two counters (first and second statistical moments for link utilization)
- Transmit only the estimate to the controller, instead of high-speed data flow
- Could be used to estimate the risk of congestion





0.3s threshold $t = 0.01$	5 minute			0.3s	
	true positive	false positive	true negative	hit	miss
$t/15$	72	23	9	1184	17
$t/10$	69	11	24	1180	21
$t/5$	49	6	49	1088	113

Monitoring support in UNIFY APIs



SP-DevOps reference-point components:

Set/GetExecutionState
Verify
GetPerformanceValues
Notify

Conclusion

- Scalability of monitoring could be improved by
  - using monitoring buses
  - adopting split-architecture principles for certain monitoring functions
- More flexible dataplane programming capabilities required for advanced monitoring
- More expressive programming capabilities required for better integrating monitoring with orchestration and controllers

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement no. 619609 - the UNIFY project. The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document

The authors acknowledge contributions from Per Kreuger (SICS Swedish ICT) and Bertrand Pechenot (Acreo Swedish ICT).