Challenges for flow-based management - implications from draft-unify-nfvrg-devops

Catalin Meirosu
Ericsson Research

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

draft-unify-nfvrg-devops

- Purpose of the document: open discussion in NFVRG on:
  - a set of principles that are relevant for applying DevOps ideas to managing software-defined telecom network infrastructures
  - challenges related to developing tools, interfaces and protocols that would support these principles and leverage standard APIs for simplifying operations tasks
- Challenge areas
  - **Stability of the software-defined infrastructure versus continuous changes**
  - Consistency, Availability and Partitioning trade-offs
  - **Observability: scalability, distribution, automation**
  - Verification: when to do, what to check, scalability
  - Troubleshooting: automated workflows
  - Identification and definition of performance metrics

Co-authors

- C. Meirosu – Ericsson
- A. Manzalini – Telecom Italia
- J. Kim – Deutsche Telekom
- R. Steinert – SICS
- S. Sharma – iMinds
- G. Marchetto – Politecnico di Torino
- I. Pappafili – OTE
- K. Pentikousis – EICT
- S. Wright – AT&T

The UNIFY project in a nutshell

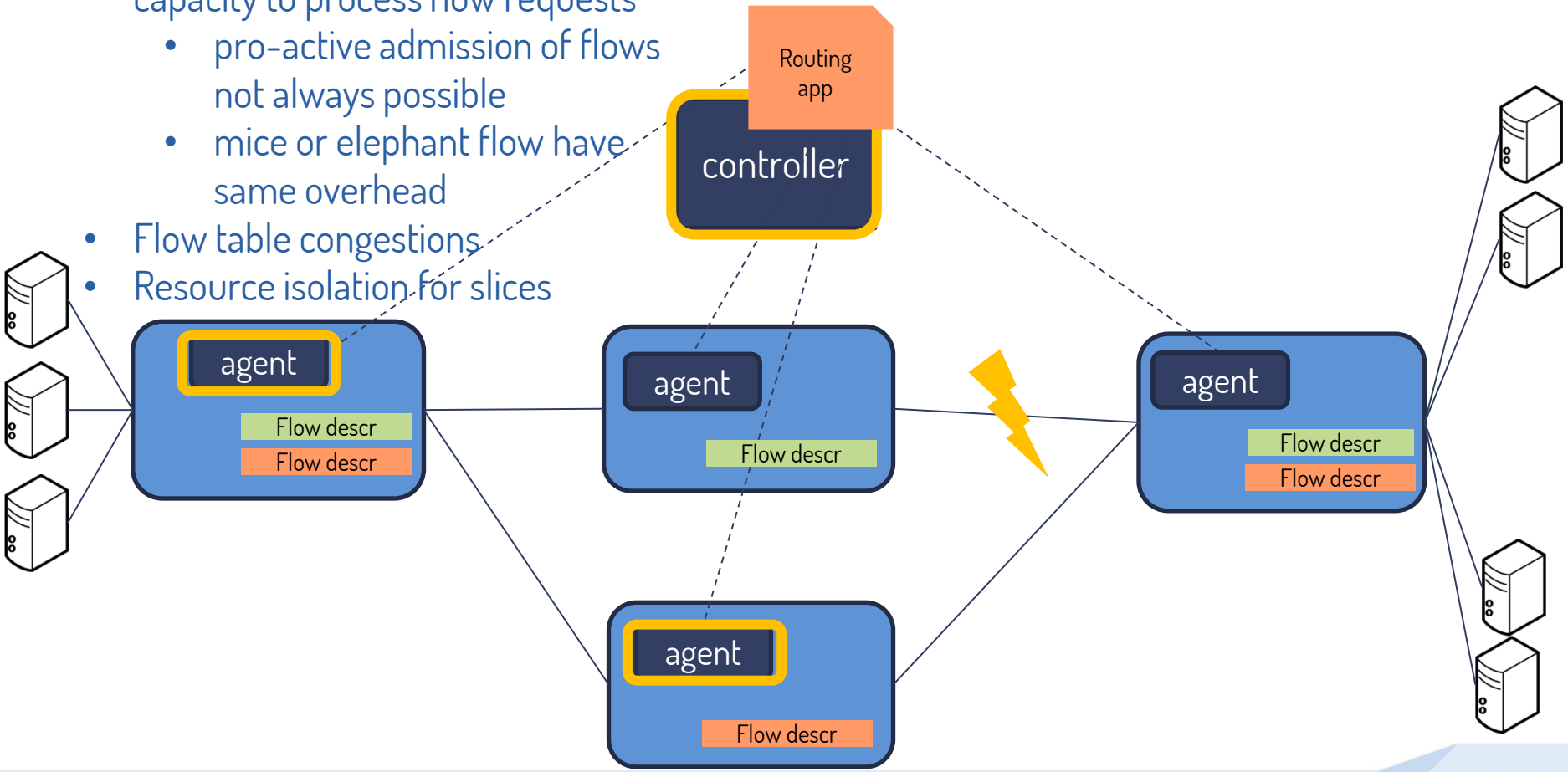
- Increase velocity of service introduction
- Unified network-cloud programming abstraction: orchestration and generic processing
- Novel observability and verification features

Types of flow descriptors

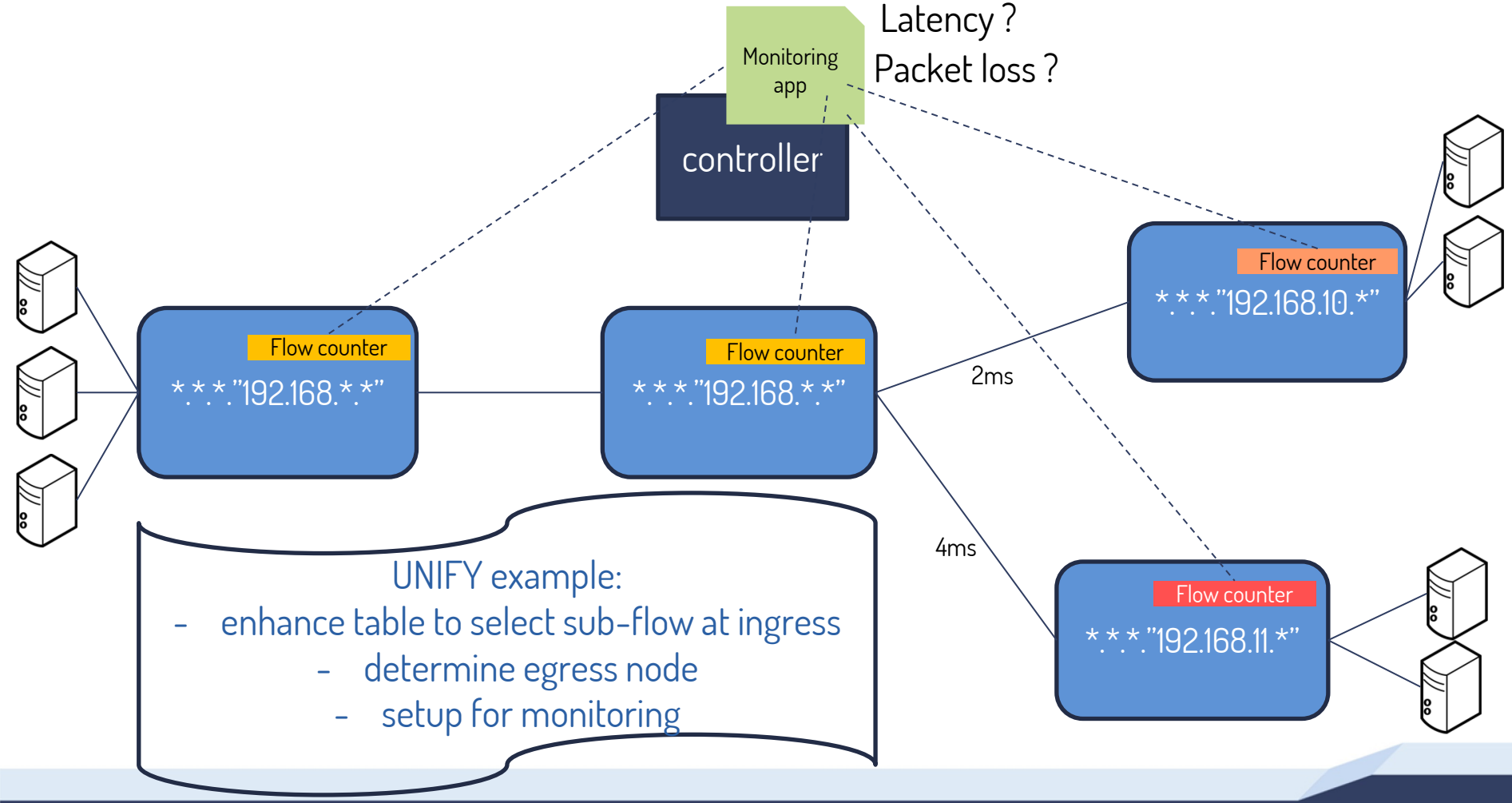
- Individual flows
  - All fields of the flow descriptor contain specific values uniquely identifying one particular flow
  - Advantage: precise visibility
  - Disadvantage: scalability (linear increase of forwarding table space)
- Aggregated flows
  - Flow descriptor contains aggregation descriptors
    - OpenFlow: *
    - CIDR: /number
  - Advantage: scalability (sub-linear increase of forwarding table space)
  - Disadvantage: loss of visibility

Infrastructure stability: flow admission control

- Controller and Agent have limited capacity to process flow requests
  - pro-active admission of flows not always possible
  - mice or elephant flow have same overhead
- Flow table congestions
- Resource isolation for slices

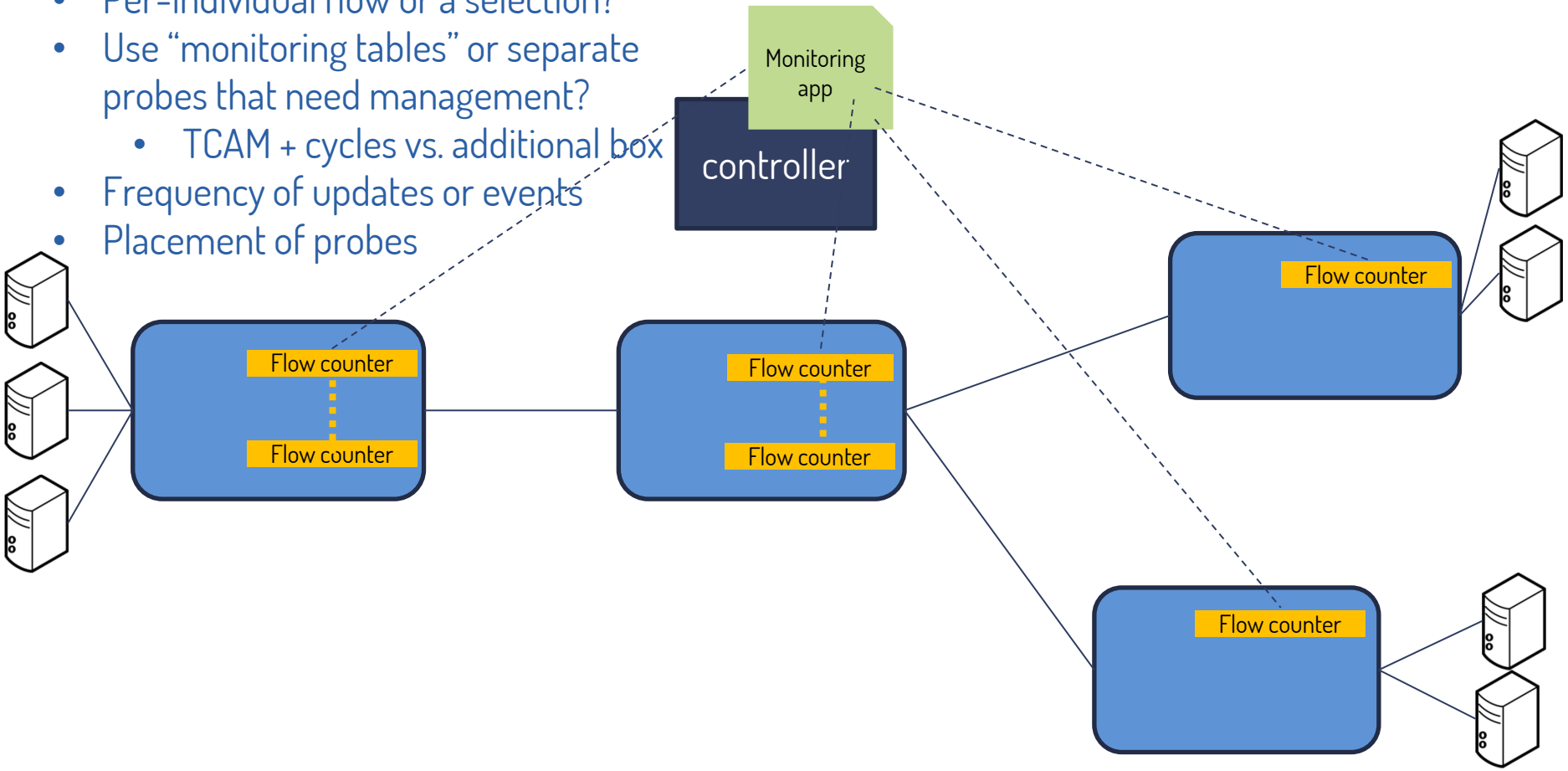


Observability: monitoring aggregated flows



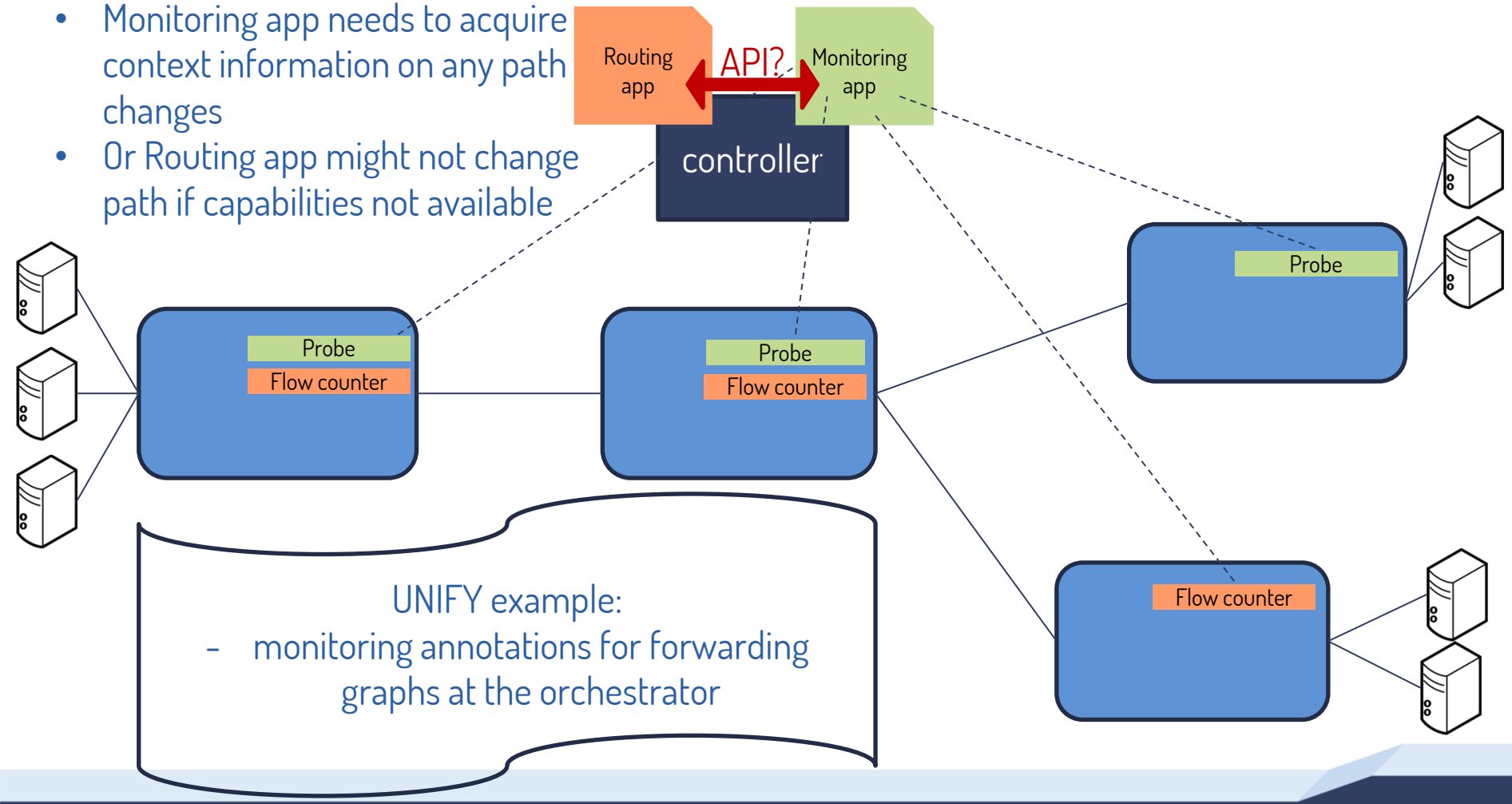
Observability: scalability challenges

- Per-individual flow or a selection?
- Use "monitoring tables" or separate probes that need management?
  - TCAM + cycles vs. additional box
- Frequency of updates or events
- Placement of probes



Observability: consistent deployment need

- Monitoring app needs to acquire context information on any path changes
- Or Routing app might not change path if capabilities not available



Observability: towards monitoring languages 1/2: current state

- Dataplane: simple incremental counters associated to flows
  - atomic operations (OpenFlow): counter automatically "moves" with flow when control plane decides migration
  - split operations (SNMP): counter no longer updated when flow moved, management needs to discover new flow path to fetch counter
- Yang-based data modelling
  - TWAMP configuration management models
  - LIME draft Yang model for performance management
  - PSAMP configuration Yang data model (RFC6728)
- OGF network monitoring schema: higher level, but not used outside academia
- Chef and Puppet templates popular with DevOps

Observability: towards monitoring languages 2/2: selected open research areas

- Data plane-level monitoring constructs beyond incremental counters:
  - "lossy" counters
  - statistical counters. UNIFY example: mean and variance
  - aggregate counters over several flows
- Enable/disable monitoring of a specific flow, including functionality at the data plane level
  - "disable counter for flow X.Y.Z.T". UNIFY example: monitoring function with statistical counter dataplane component instantiated upon provisioning of the forwarding graph it is associated with
- Need for an expressive and extensible language able to:
  - define monitoring fabrics, with aggregations and triggers
  - easily extendable to describe parameters associated to self-adaptive monitoring functions
    - accuracy versus overhead trade-offs
    - transparent adaptation of measurement frequency to situation in the infrastructure
  - support event-condition-action rules

Conclusion

- draft-unify-nfvrg-devops defines a series of challenges, relevant also for flow-based management
- Observability is a particularly challenging area, and more work is needed on languages and capabilities for flow monitoring
- The UNIFY project addresses flow management challenges in software-defined infrastructure
  - For more details, the D4.x deliverables and M4.1 are available at https://www.fp7-unify.eu/index.php/results.html#Deliverables

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