From Slicing to Dynamic Resource Control

Dirk Trossen*

InterDigital Europe,
London, UK

dirk.trossen@InterDigital.com

Rui L. Aguiar*

Instituto de Telecomunicações / DETI
Universidade de Aveiro, Portugal

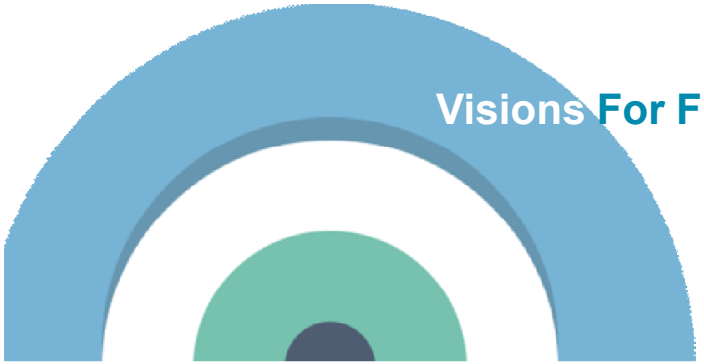
ruilaa@ua.pt

Artur Hecker*

Huawei ERC,
Munich, Germany

artur.hecker@huawei.com

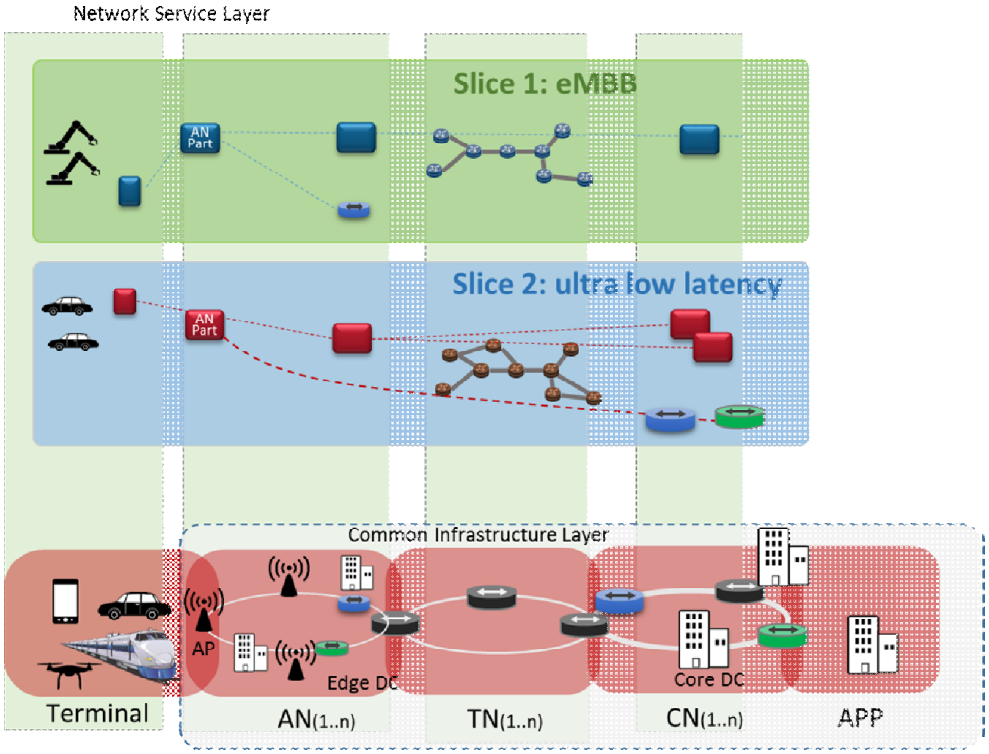
**Without any particular author order*

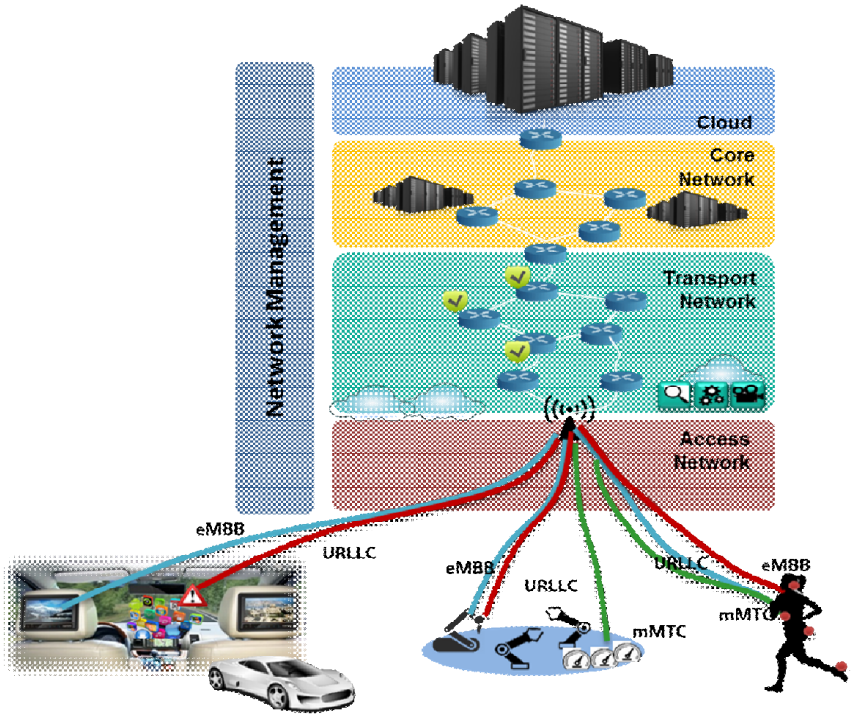


Visions For Future Communications Summit | Lisbon, Portugal, Oct 23-34, 2017

5G Slicing: Common Understanding

Definition: 5G E2E Network Slicing is a concept for running multiple logical networks
(which could be customized and with guaranteed SLA)
as virtually independent business operations on a common physical infrastructure.





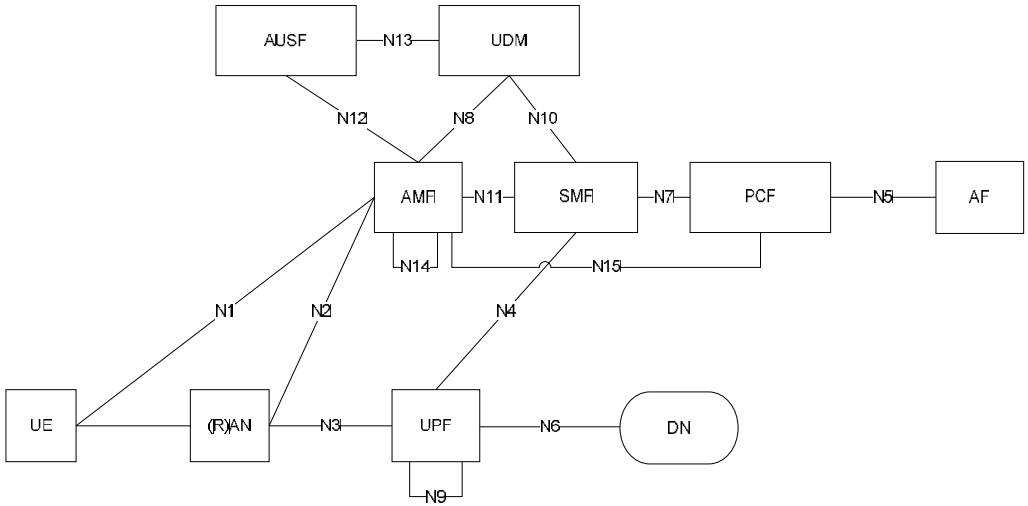
3GPP: Slicing and atomic functions in standardization

The general direction has been **acknowledged and standardized by 3GPP**

From 3GPP TR23.799

Key issue 1: Support of network slicing
Key issue 2: QoS framework
Key Issue 3: Mobility management framework
Key issue 4: Session management
Key issue 5: Enabling (re)selection of efficient user plane paths
Key issue 6: Support for session and service continuity
Key issue 7: Network function granularity and interactions between them
Key issue 8: Next Generation core and access - functional division and interface
Key Issue 9: 3GPP architecture impacts to support network capability exposure
Key issue 10: Policy Framework
Key issue 11: Charging
Key issue 12: Security framework
Key issue 13: Broadcast/Multicast Capabilities
Key Issue 14: Support for Off-Network Communication
Key Issue 15: NextGen core support for IMS
Key Issue 16: 3GPP system aspects to support the connectivity of remote UEs via relay UEs
Key Issue 17: 3GPP architecture impacts to support network discovery and selection
Key Issue 18: Interworking and Migration
Key Issue 19: Architecture impacts when using virtual environments
Key issue 20: Traffic Steering, Switching and Splitting between 3GPP and non-3GPP Accesses
Key Issue 21: Minimal connectivity within extreme rural deployments

In red: priority key issues

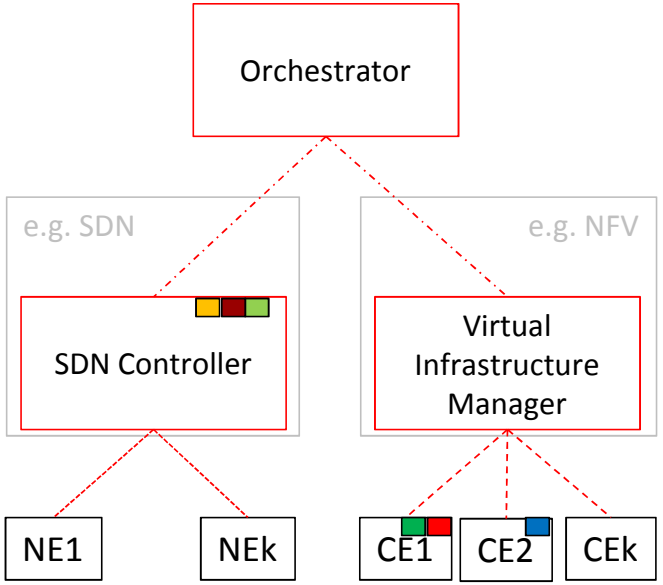


3GPP Current Working Model **TS23.501**
Modularization as key design element
Also: ***Service-Based Architecture***

NGMN: slice templates as representations of long term business contracts

State of the art in network slicing and softwarization

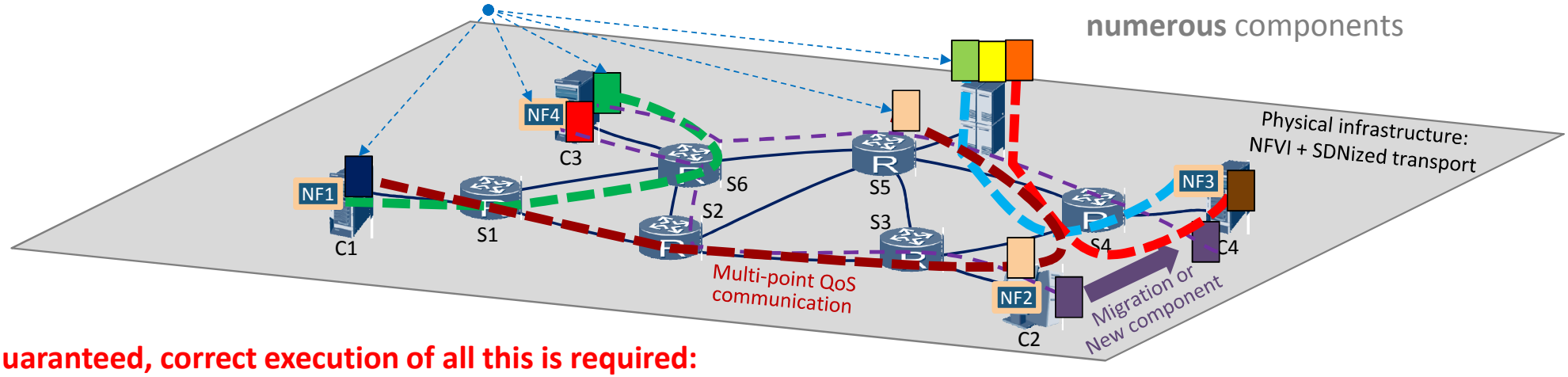
- Current approach: cross-domain management++
  - Through novel and more complex management functions
  - Architecture: OSS/BSS gets extended with new functions for slice creation, deletion, monitoring
  - Those functions get implemented by extending and translating between different information models in AN/RAN, TN, Core, etc.
- Doubts with that approach:
  - How to do E2E? Management traditionally is bad with heterogeneity
  - Far away from the devices, from network events
  - Far away from networking – too centralized, too high level, too IT ☹
  - Results in very static slicing with pre-provisioning:
    maybe OK for less than 10 slices



Slicing: from HW Composition to SW Composition
Plethora of modules/paths - must work correctly any time

Paths and end-points are part of the potentially critical internal network function realization

Continuous **creation** and **removal** of **numerous** components



Guaranteed, correct execution of all this is required:

- **Correct local execution of hundreds of thousands of components**
  - Shared with other executed components, yet guaranteed – scheduling
- **Correct, guaranteed interconnection of components in spite of contention**
- **Timely control of all this**
  - For creation/migration, path QoS, execution guarantees, checks, quotas, …
- **Correct function of the executing environment**
  - Including all resources (compute, network)
  - … and their control systems

5G requires more than best effort because of its own realization

Dynamic slicing: adapt the slice in runtime
Execution control

- Static slicing is a waste of resources
  - Differences in usage between day and night
  - Differences in geographical usage
  - Green Networking
  - Service popularity/success difficult to predict
    - Example: SMS, Pokémon Go
- **Dynamic slicing** allows to adapt a slice
  - Like in the Cloud/DC: scale in / scale out
  - Dynamic resource assignment
  - Dynamic scheduling depending on real usage

Static slicing is like a **bus lane**

Software Networks: A metaphor

Before

Ready to use

- Can type a text straightaway

Hardware defines the service (1 service)

- Hardware limitations = service limits

NFV/SDN

After

Not ready to use

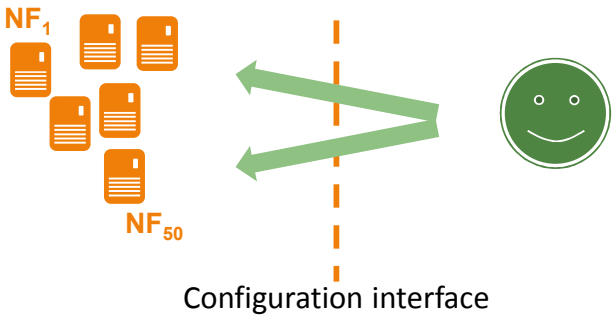
- **Need to install and start MS Word first (mgmt)**
- **Need to execute MS Word (runtime control)**
  - **Runtime resource allocation (CPU, memory, disk)**
  - **In parallel to other applications**

Software defines the service (N services)

- Hardware limitations = service **quality** limits

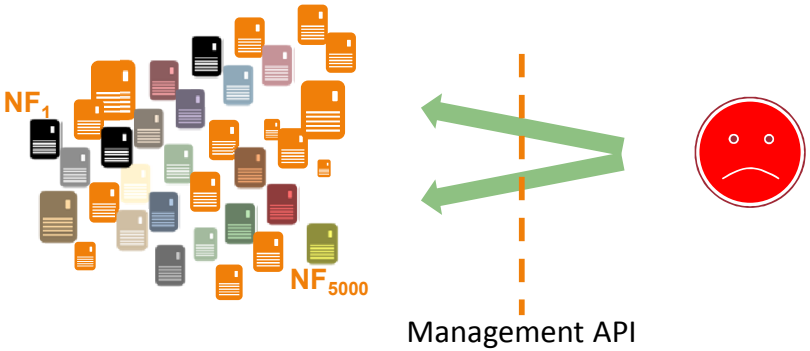
On the expected dynamics

Today



Running on distinct, well-tested appliances
Well-tested, unshared, stable, low failure rates

Sliced 5G

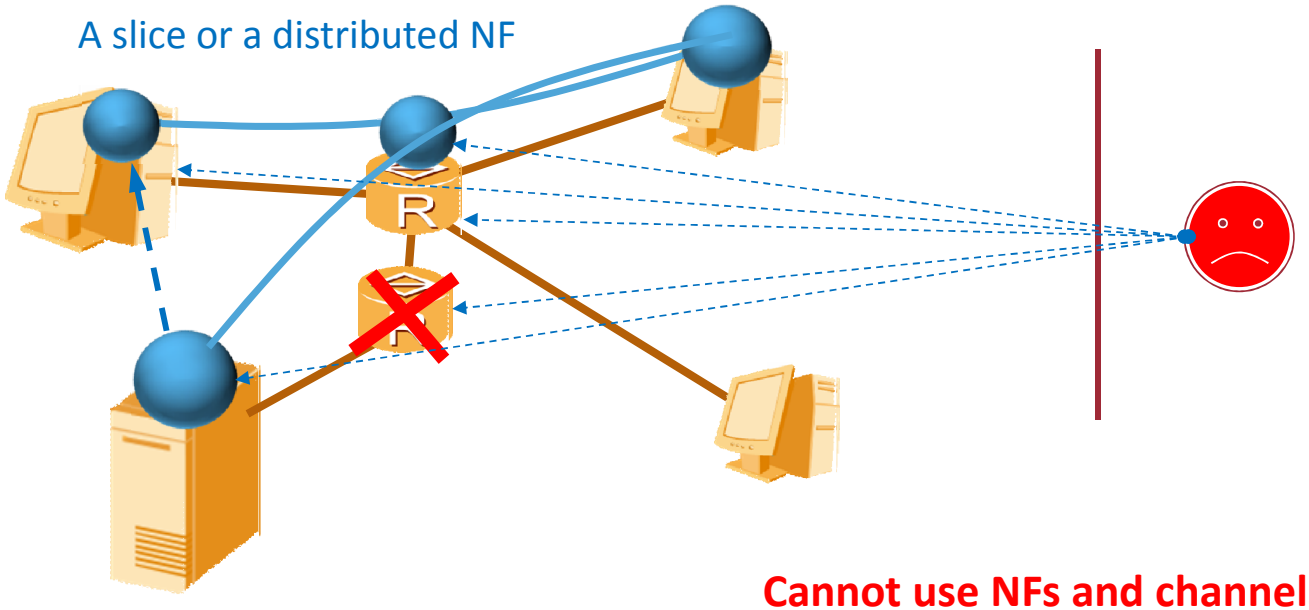


Running on a shared pool of COTS appliances
Huge numbers, mobility of network elements, uncertain outcome, sharing, moderate failure rates: too high load

CONCLUSION

Management cannot work as the foundation of the NF configuration.
We need automation, i.e. **control**

Software Networks: on runtime coordination



Cannot use NFs and channels independently
Physical failures will result in NF reconfiguration!

CONCLUSION

Pushing such treatment to management is possible but too slow
Need to keep it at resource layer, i.e. akin to **control**

On the expected scales
Number of modules, number of paths

Number of modules (#NFI) =

#Slices X (#NF types per slice) X (#NF instances)

~1000 X ~10 X ~10 = 10^5 of NF Instances running in the network

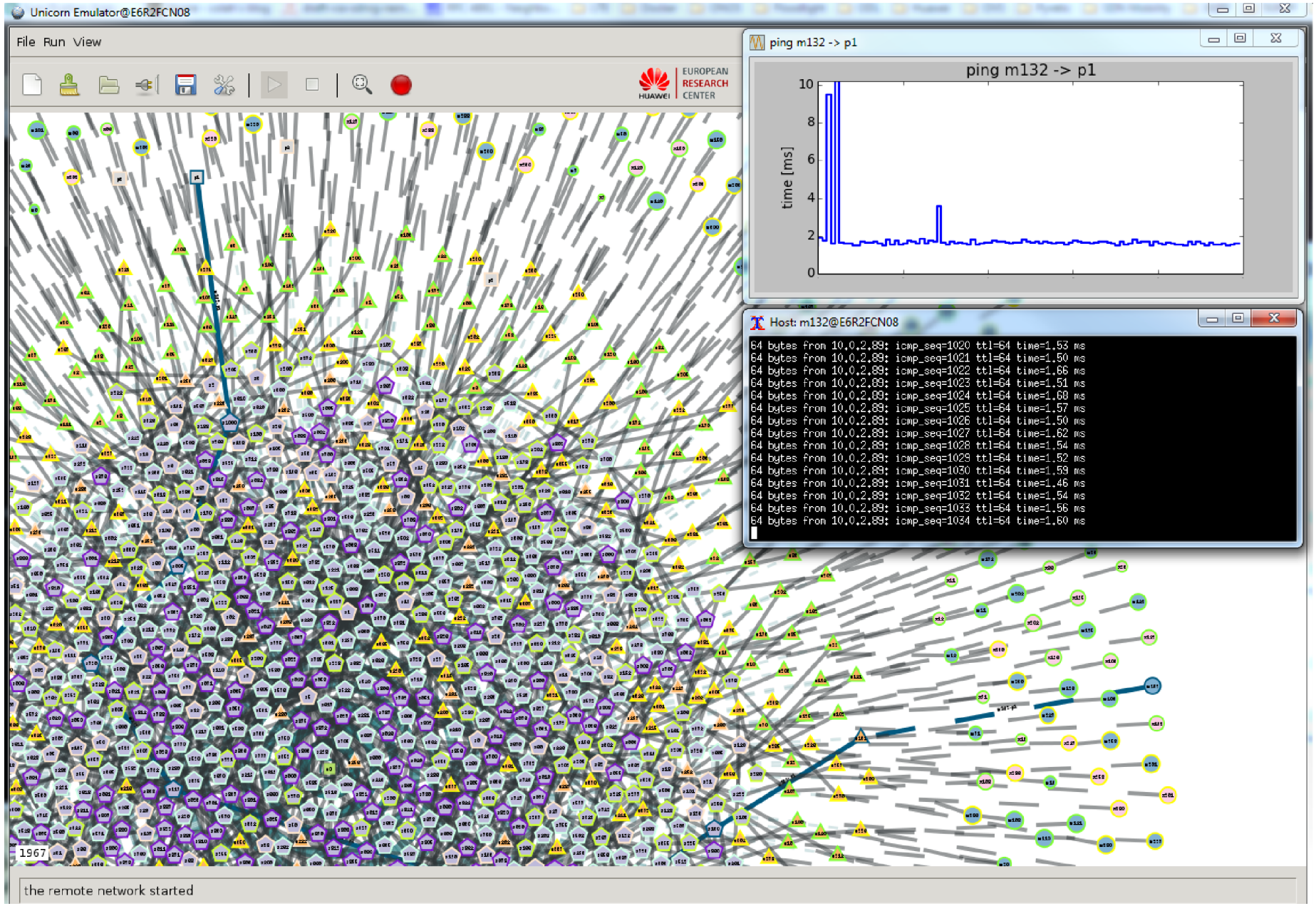
e.g. according to NGMN, operators

cf. 3GPP NextGen SA2, RAN3

Example. Depends on type and load.
e.g. how to serve billions of IoT/M2M devices?

Number of Paths between modules: $\in [10^6 ; 10^{10}]$ worst case scales in $O(\#NFI^2)$

CONCLUSION We need a lot of automation, i.e. **control**!



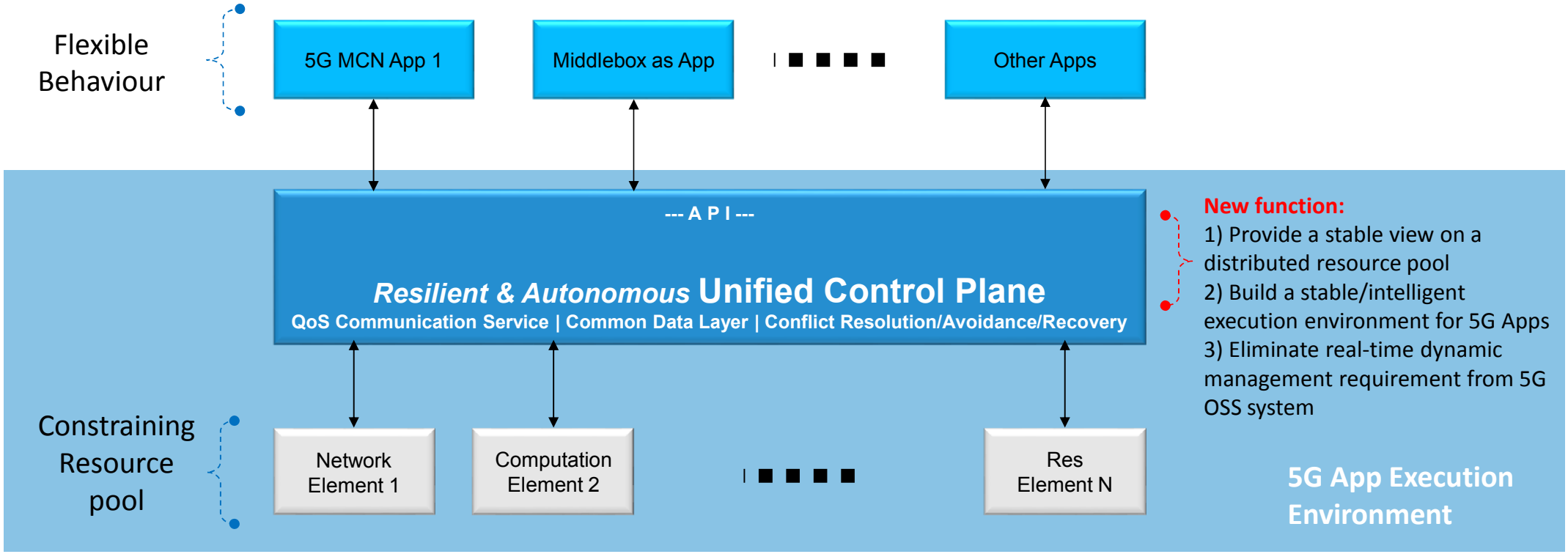
HUAWEI | EUROPEAN RESEARCH CENTER

We argue that Control is the right approach to address the softwarization challenges

Controlling Slices and Sliced Networked Systems
Some Solution Elements

- Move from Design-Time approach to **Runtime approach**
  - Ownership through control
  - Growing from within
- Build for **High Dynamics**
  - Move from dynamic loads to dynamic presence (churn) and to dynamic topologies
  - Support for scale in and scale out, including for your own slice support systems
- Embrace **Distributed Software systems principles**
  - Generic Interfaces and Self-describing Objects, instead of predefined information models
  - Value Scalability over Optimality
  - Conflict Avoidance and Resolution, move away from Sequential Models
  - Prepare for Loose and Eventual Consistency Models, instead of presumed full correctness
  - Dynamic Resource Management: Garbage Collection for Networks

Runtime Control
Autonomic in setup and distributed by default



Compare to IETF ANIMA, IETF SFC

Conclusion

- Slicing and network softwarization are new challenges in networking
  - So far, only orchestration and management are properly addressed
- The **execution and runtime problems** are mostly ignored
- Their solution requires **new fundamental research**
  - **Resilient Integrated Control of an unstable set of distributed resources of different nature**
  - **Distributed Scheduling on top of such a resource set**
  - **Conflict Avoidance and Resolution**
- Needs:
  - Least common denominator for programmable networks of the future
  - Distributed Network wide primitives with good runtime scheduling for jobs and flows

THANK YOU