Autonomic Slicing

Background and Possible extension to Anima

31st March 2017

draft-galis-anima-autonomic-slice-networking-02

Prof. Alex Galis
a.gais@ucl.ac.uk; http://www.ee.ucl.ac.uk/~agalis/

University College London,
Department of Electronic & Electrical Engineering
Torrington Place
London WC1E 7JE
United Kingdom

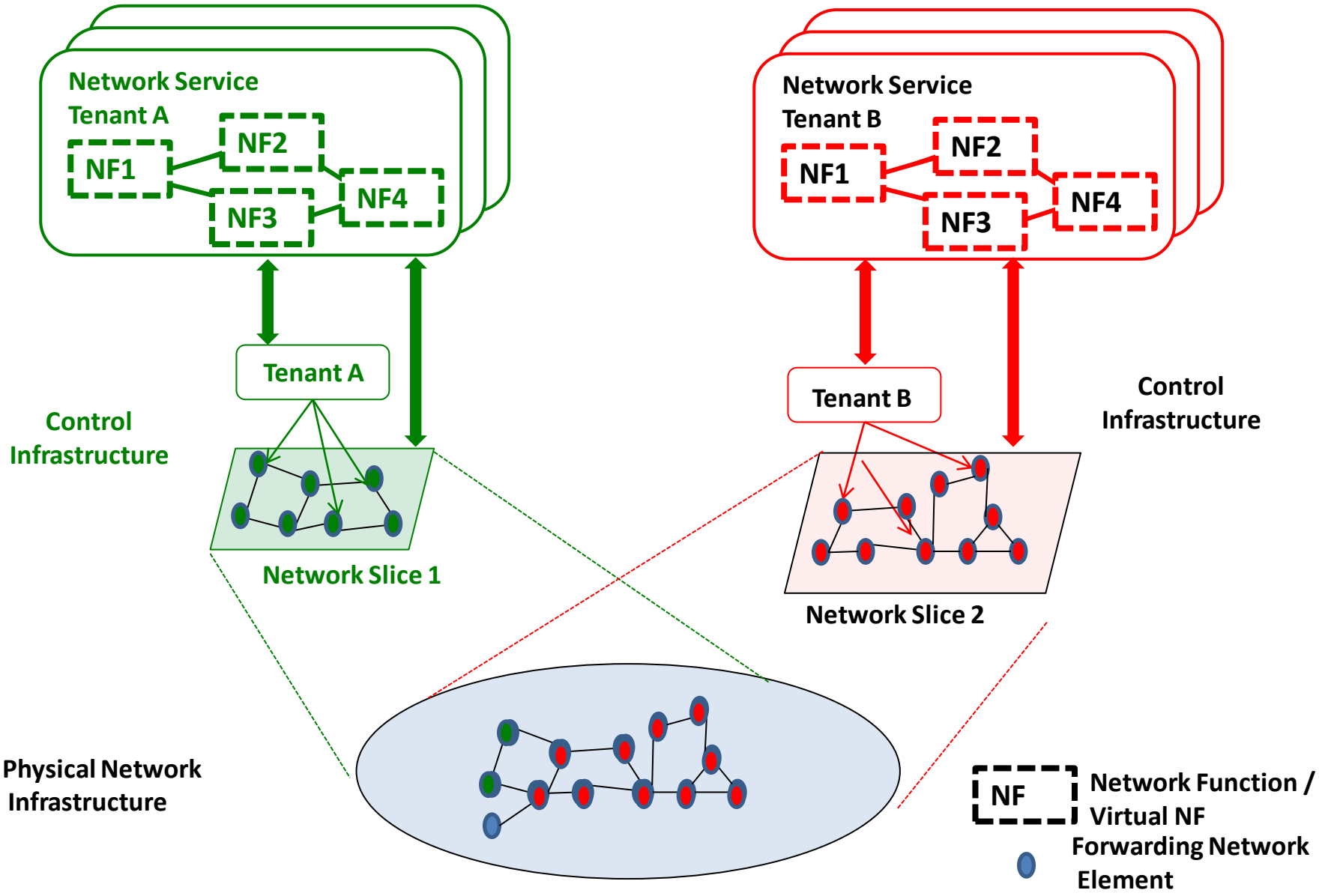
Leo Liubing
leo.liubing@huawei.com

Huawei Technologies
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China

What is Network Slicing?

- A brief re-call of last presentation
- Network Slicing (NS) definition
  - ITU-T/5G definition
- Early NS
  - Physical partition of networks (e.g. VPN, Overlay, Virtual Networks)
- Modern NS: Slice as a union of subsets of resources & NFVs at a given time
  - Logical partition of a network device; and
  - Virtual resources created in NFV; and
  - Logical partition of a network functions (of the data, control, management plane)
  - NS Manager with NS capability exposure

Slice as a union of subsets of resources & NFs



What are the NS Characteristics?

- Service customized Network Slices + Smart Network Fabric for coordinating/orchestration, control of network resource.
- Concurrent deployment of multiple logical, self-contained and independent, shared or partitioned networks on a common infrastructure platform.
- Supports dynamic multi-service support, multi-tenancy and the integration means for vertical market players.
- Separation of network functions simplifies.
  - the provisioning of services,
  - manageability of networks and
  - integration and operational challenges especially for supporting communication services.

Why are we here in Anima?

- Network Slicing is becoming an IETF work
- Automaticity is an intrinsic requirement for NS
  - Fits into Anima's scope (with reasonable extension, see later discussion)
  - A centralized approach (i.e. SDN like) may not be applicable

New Features:

- Autonomic functionality per network slices
- NS transforms the networking perspective by
  - Abstracting, Isolating, Orchestrating,
  - Separating logical network behaviors from the underlying physical network resources.

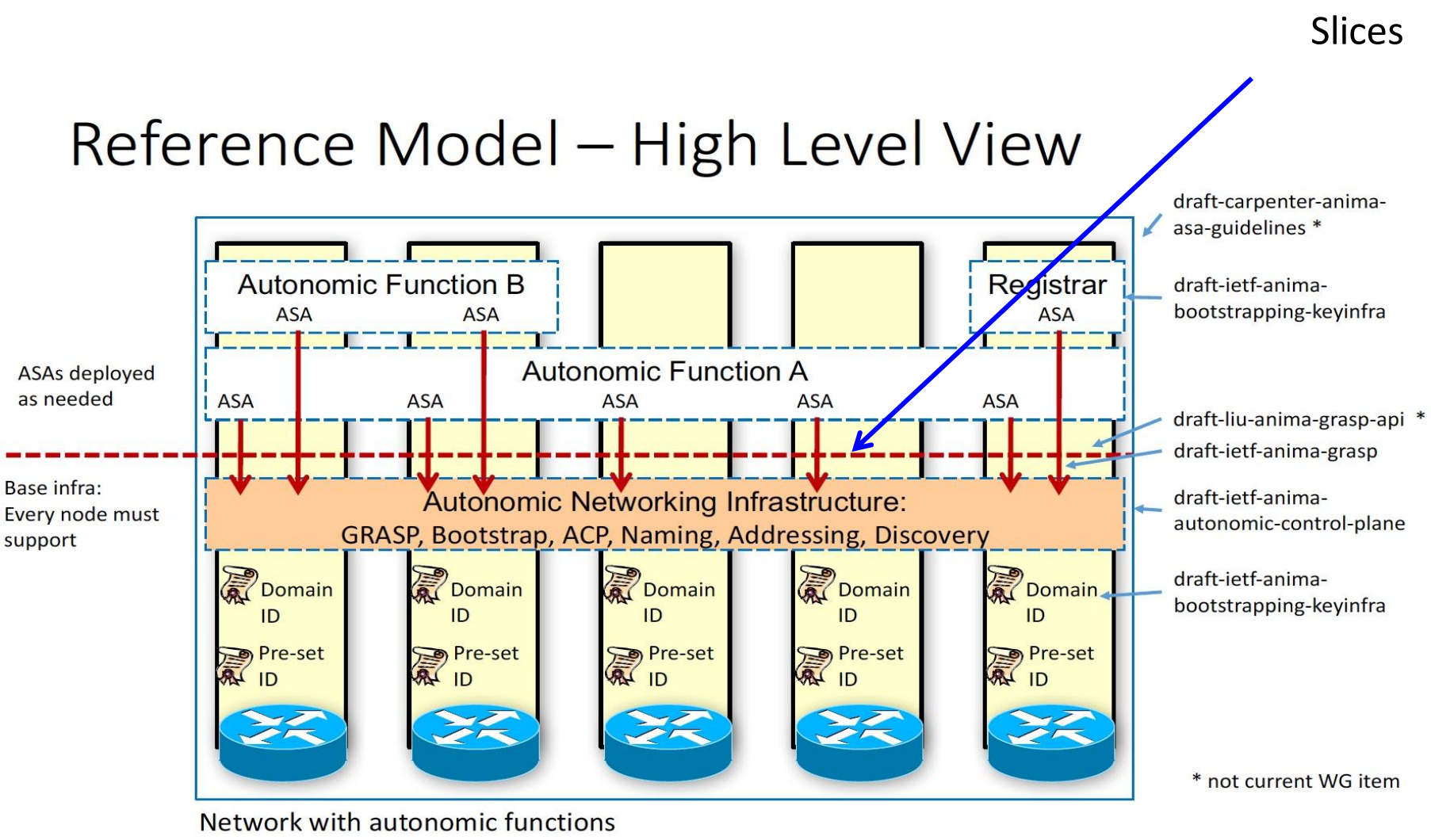
Impact to Anima

- Virtualization environment
  - Modern Network Slicing requires virtualization
  - Potential extension to current reference model and protocols to fit into virtualization environment
- Higher degree signaling interaction
  - Slice protocol characterization – operation, (de)composition, creation, deletion must also work completely automatically
  - These are crucial “commands” than normal signaling interaction
  - More reliable and stationary designs might be needed
- Specific management requirements
  - NS is based on principles of Self-management: self-configuration, self-composition, self-monitoring, self-optimisation, self-elasticity
  - Sophisticated info/data model might be needed for communicating between ANs to fulfill the self-x tasks

Anima fits into virtualization environment

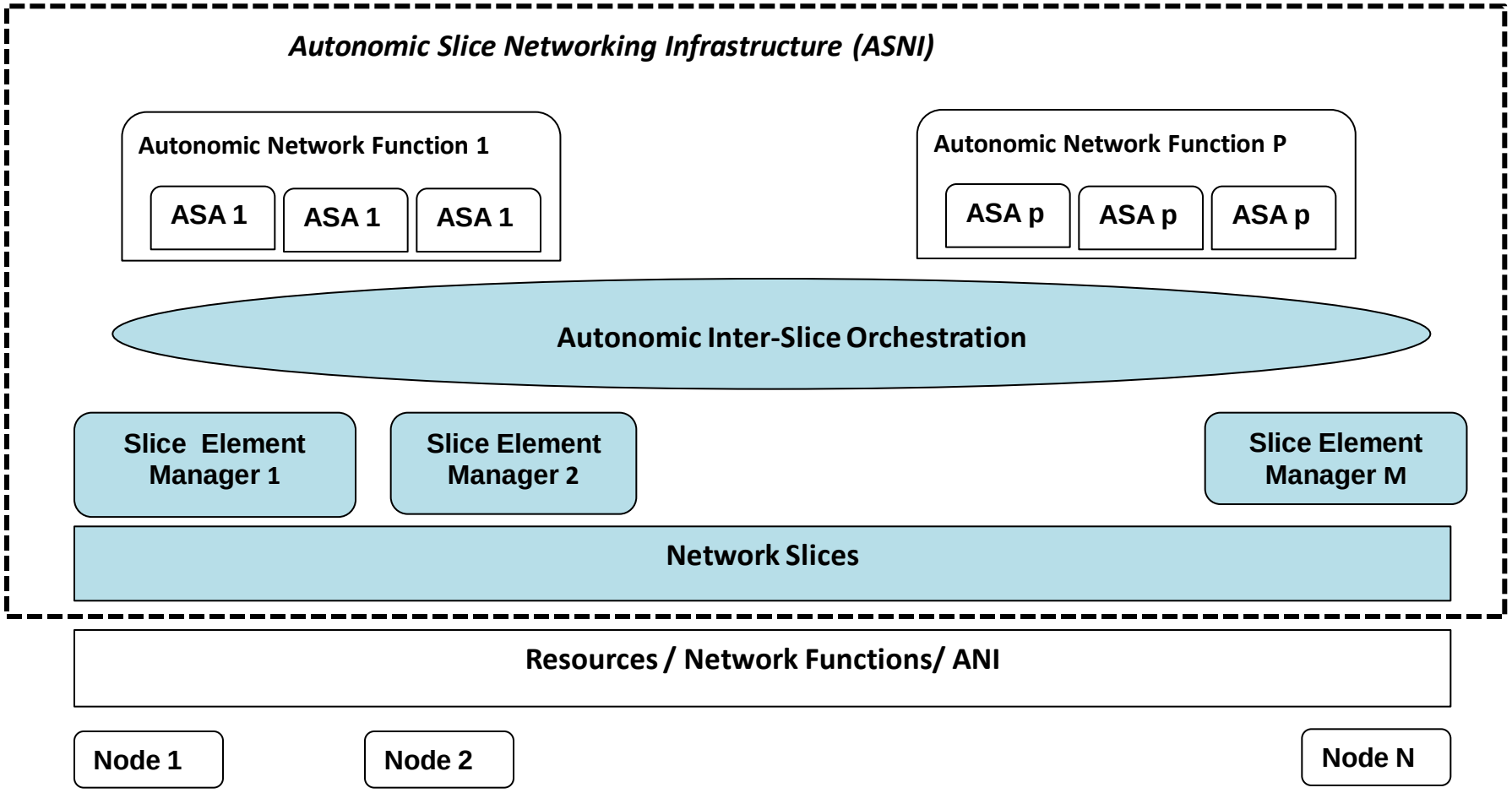
- Anima reference model
  - Autonomic **Node** -> Autonomic **Managed Object**
- Bootstrap
  - Bootstrap of a virtual object (e.g. a slice) needed?
- ACP
  - The management channel is among physical nodes, or among virtual overlay(s)?
- GRASP
  - Supporting virtual objects communication?

NS Automaticity fits for Anima's approach

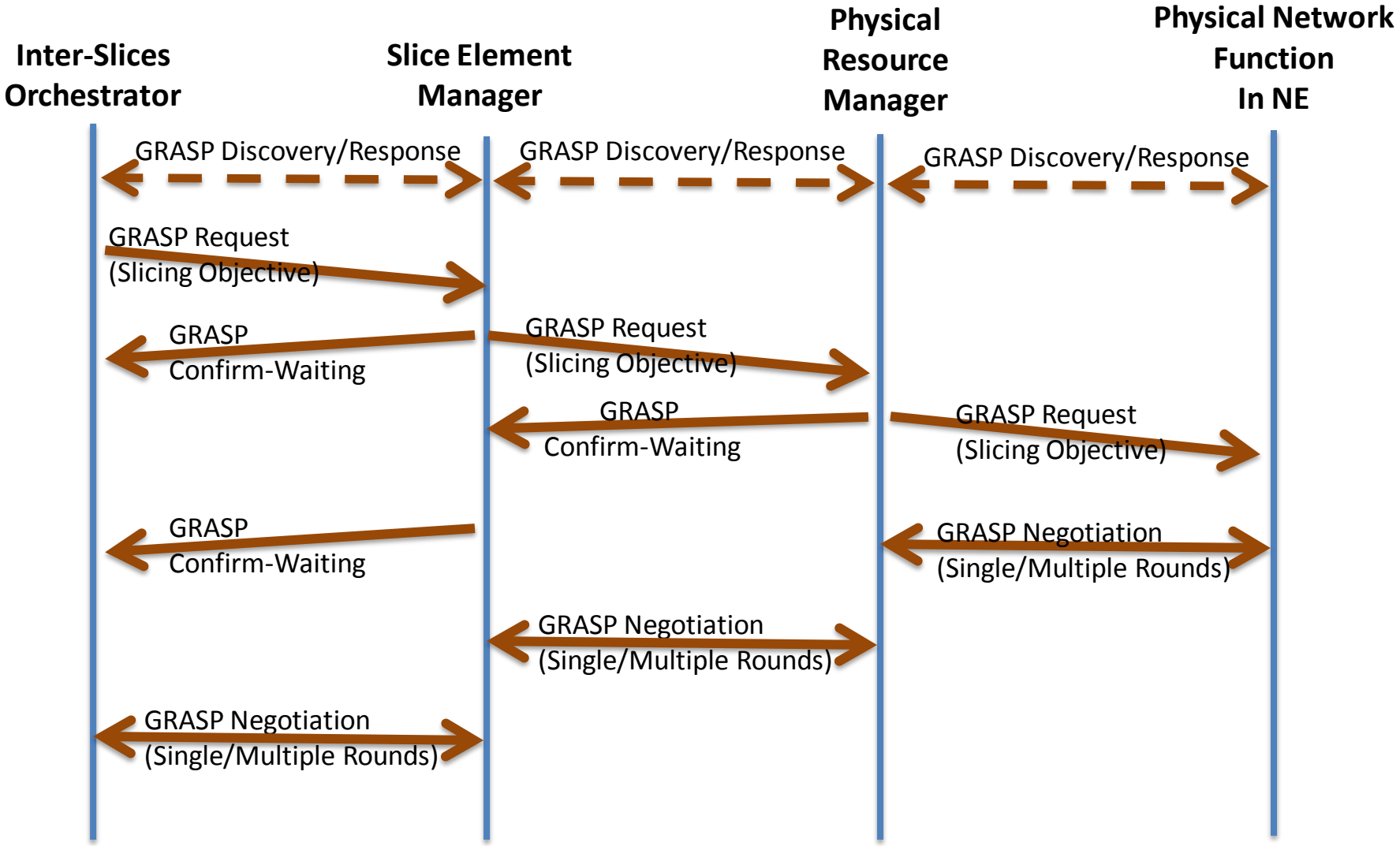


NS Automaticity fits for Anima's approach

- **"Autonomic Slice Networking Infrastructure" (ASNI)** - It consists of a number of autonomic nodes resources, which interact directly with each other. Those autonomic nodes resources provide a common set of capabilities across a network slices. The ASNI provides functions like naming, addressing, negotiation, synchronization, discovery and messaging.
- **Autonomic network functions** typically span several slices in the network. The atomic entities of an autonomic function are called the "**Autonomic Service Agents" (ASA**), which are instantiated on slices.



NS signaling using GRASP



Hints:

- **Cascaded GRASP negotiation**
- **Objective definition among different negotiation pairs should be different**

Inter-Slices Orchestration refers to the system functions

- automated and autonomically co-ordination of network functions in slices.

- autonomically coordinate the slices lifecycle and all the components that are part of the slice (i.e. Service Instances, Network Slice Instances, Resources, Capabilities exposure) to ensure an optimized allocation of the necessary resources across the network.

- coordinate a number of interrelated resources, often distributed across a number of subordinate domains, and to assure transactional integrity as part of the process.

- autonomically control of slice life cycle management, including concatenation of slices in each segment of the infrastructure including the data pane, the control plane, and the management plane.

- autonomically coordinate and trigger of slice elasticity and placement of logical resources in slices.

- coordinates and (re)-configure logical resources in the slice by taking over the control of all the virtualized network functions assigned to the slice.

Slice Element Manager Functions

Slice autonomic management is driven by Slice Element Managers ; there are five categories of operations:

- **Creating a network slice**:
  - ✓ Receive a network slice resource description request, upon successful negotiation with SSA allocate resource for it.
- **Shrink/Expand slice network**
  - ✓ Dynamically alter resource requirements for a running slice network according service load.
- **(Re-)Configure slice network**
  - ✓ The slice management user deploys a user level service into the slice. The slice control takes over the control of all the virtualized network functions and network programmability functions assigned to the slice, and (re-)configure them as appropriate to provide the end-to-end service.
- **Destroy slice network**
  - ✓ Recycle all resource from the infrastructure.
- **Self-X slice management and operation**
  - ✓ namely self-configuration, self-composition, self-monitoring, self-optimisation, self-elasticity would be carried out as part of new slice protocols.

GRASP Requirements & Extensions (examples)

- **Discovery of SEMs**
  - A process by which an one SEM discovers peers according to a specific discovery objective. The discovered SEMs peers may later be used as negotiation counterparts or as sources of other coordination activities.
- **Negotiation between SEMs**
  - A process by which two SEMs interact to agree on slice logical resource settings that best satisfy the objectives of both SEMs.
- **Synchronization between SEMs**
  - A process by which Orchestrator and SEMs interact to receive the current state of capability exposure values used at a given time in other SEM. This is a special case of negotiation in which information is sent but the SEM or Orchestrator do not request their peers to change configuration settings.
- **Self configuration of SEMs**
  - A process by which Orchestrator and SEMs interact to receive the current state of capability exposure values used at a given time in other SEM . This is a special case of synchronization in which information is sent and the SEM is requesting their peers to change configuration settings.

GRASP Requirements & Extensions (examples)

- **Self optimization of SEMs**
  - A process by which Orchestrator and SEMs interact to receive the current state of capability exposure values used at a given time in other SEMs. This is a special case of configuration in which information is sent and the SEM is requesting their peers to change logical resource settings in a slice based on an optimisation criteria.
- **Mediation for slice resources**
  - A process by which two SEMs interact to agree to logically move resources between slices that best satisfy the objectives of both SEMs triggering of slice elasticity and placement of logical resources in slices. This is a special case of negotiation in which information is sent Orchestrator do request SEMs to change logical resource configuration settings.
- **Triggering and governing of elasticity**
  - A process for autonomic scaling intent configuration mechanism and resources on the slice level; it allows rapid provisioning, automatic scaling out, or in, of resources. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored slices.
- **On-demand a self-service network slicing**

Conclusion & Opportunities

- Autonomic management of slices as subnetworks in ANIMA → Management of multiple logical, self-contained and independent, shared or partitioned networks on a common infrastructure platform.

- Problem statements (i.e. ANIMA future work)
  - Anima reference model
  - Bootstrap
  - ACP
  - GRASP extensions
  - Slice autonomic operations