



UCL

Network Slicing Supported by Dynamic VIM Instantiation

Stuart Clayman
Dept of Electronic Engineering
University College London

Overview

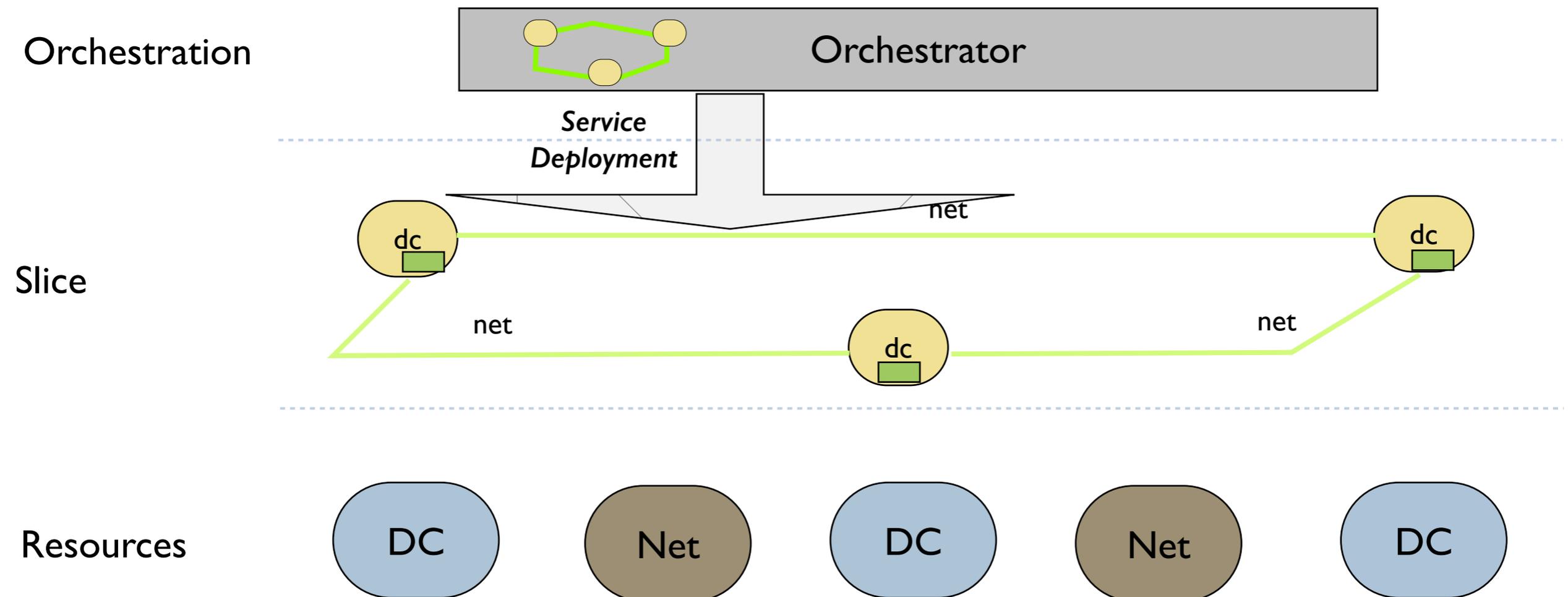
- Here we present an overview of some of the mechanisms, components, and abstractions that can be utilized in order to encompass network slicing into a bigger picture for NFV delivery.
- In particular, we make the case for Data Center (DC) slicing, as part of the full NFVI foundation, to ensure that the attributes prescribed to network slices are propagated into the DC.
- There are some scenarios in which it is important to have a separate Data Center slice within a full Network Slice.

Overview

- We make the case for creating a VIM on-demand and dynamically allocating a new VIM (Virtual Infrastructure Manager) for each slice, rather than having one for the whole DC, which can be beneficial for those scenarios.
- We show the architectural elements that are required to support such a model, as well as a set of layered abstractions using slicing elements, showing how they all fit together for service provisioning and integrate with an orchestrator.
- The following slides show the what and the how.

What is the target

- Slicing is a move towards segmentation of resources and deployment of NFV for the purpose of enhanced services and applications on a shared infrastructure.

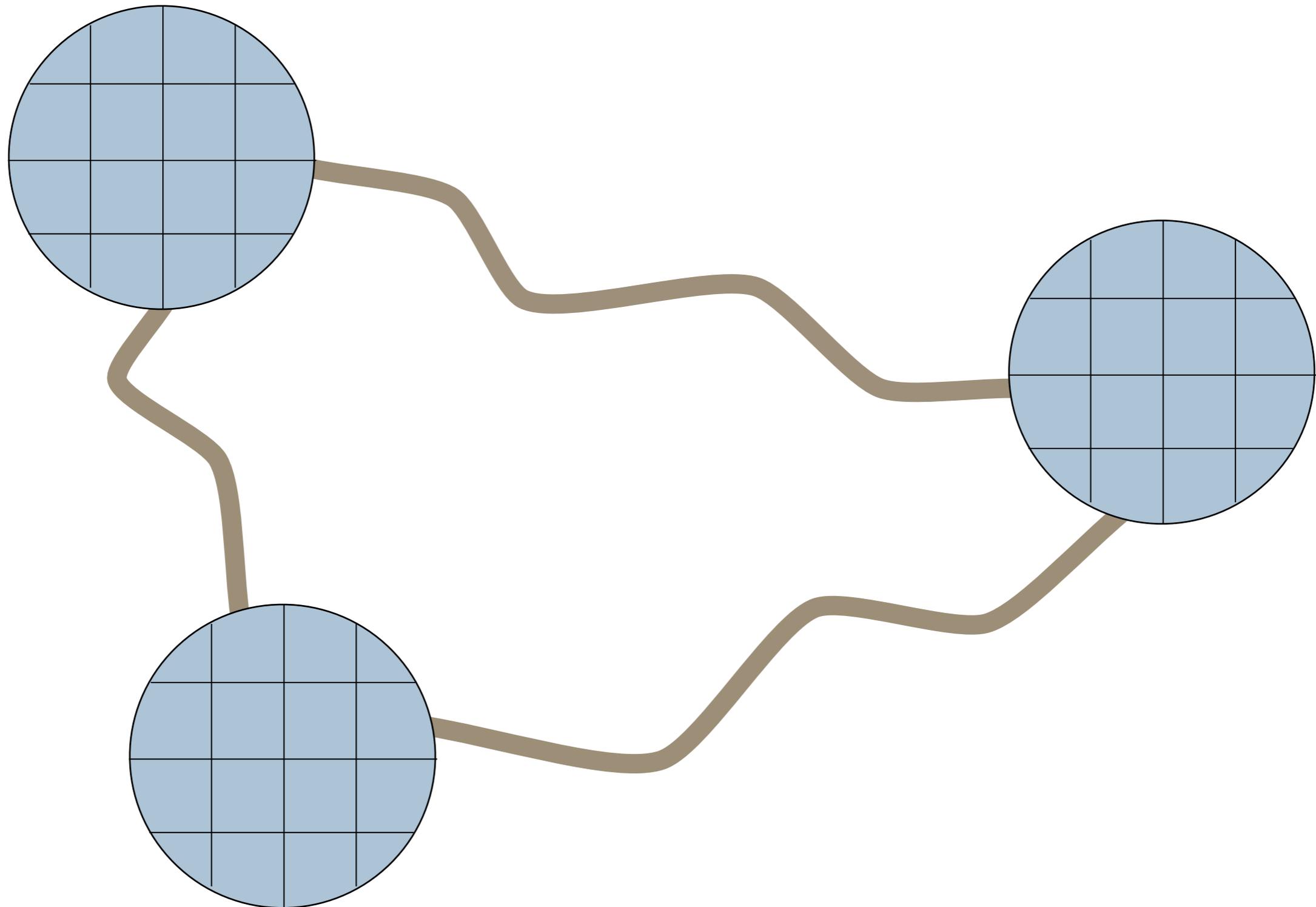


Where is the issue?

- Currently many network slicing models slice the network, but have NFV elements scattered across the DC, due to various reasons.
- Slices could considerably transform the networking perspective and enhance NFV architecture by:
 - Abstracting
 - Isolating at a sub-network level
 - Separating logical network behaviours from the underlying physical network resources
 - Allowing dynamic management of network resources by managing resource-relevant slice configuration
 - Simplifying and reducing the expenditure of operations
 - Support for rapid service provisioning
 - Support for NFV deployment

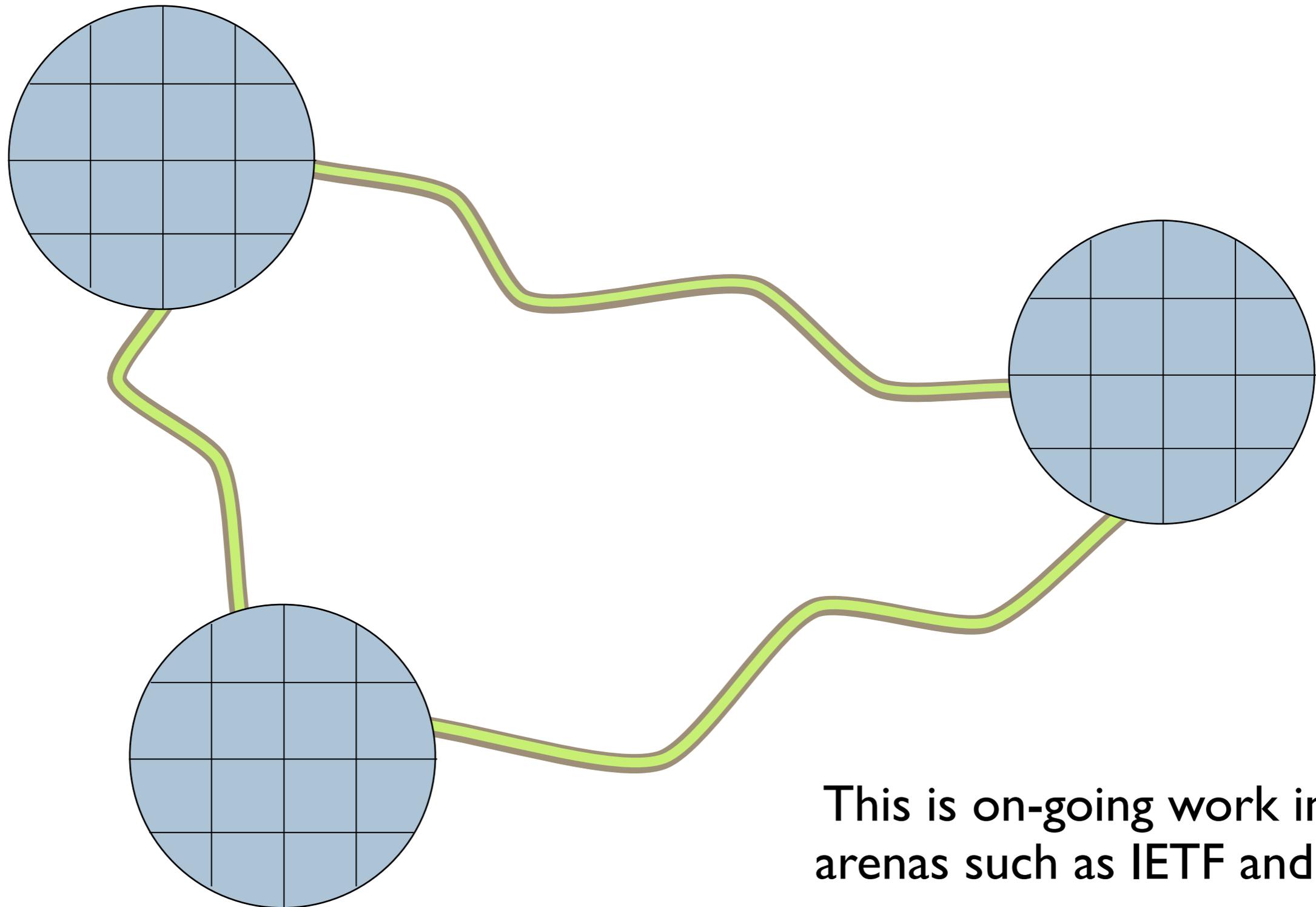
Slicing

The DCs and the networks are physically connected.



Slicing

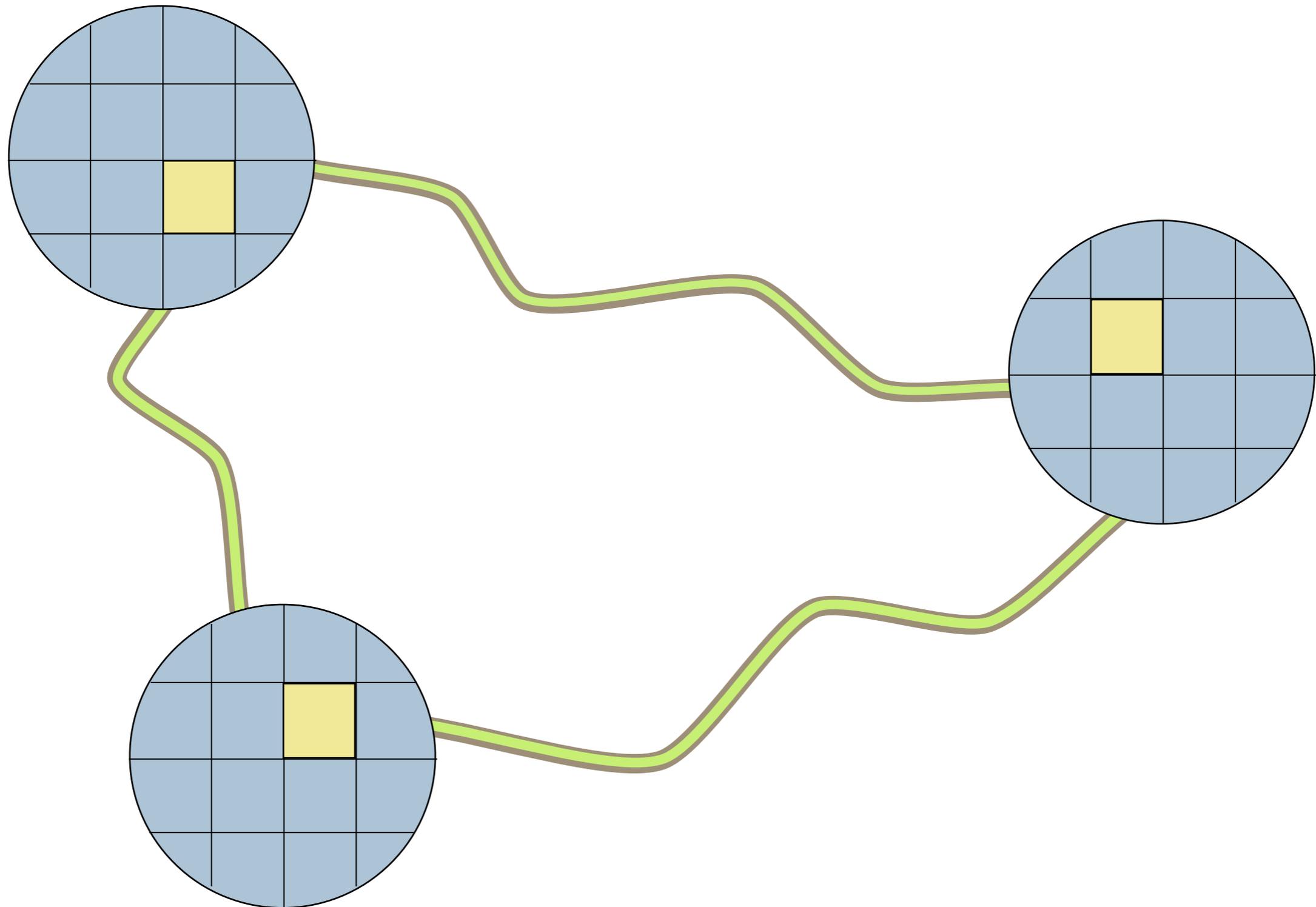
Slices can be requested from networks.



This is on-going work in many arenas such as IETF and IEEE.

Slicing

Slices should also be a feature that can be requested from DCs.



DC Slice

- A DC slice is an abstraction over the resources of a DC
 - It presents a collection of resources that look like a DC
 - It can be controlled and managed independently from any other slices
- A DC slice can be allocated at any Data Center: large centralized DCs, medium DCs, and mobile edge DCs.
- A slice is a basis for control in virtualized environments:
 - a DC slice needs to be as elastic as other elements
 - it can grow or shrink dynamically under the control of a Slice Controller

DC Slice Attributes

- For each slice there will be an on-demand VIM allocated for any kind of lower level virtualization, including Xen, kvm; or for containers such as Docker / kubernetes.
- As a consequence, this choice is not a feature pre-determined once by the DC or the provider, but can now be an option for the customer.
- The customer will have a lot more configuration options for their VIM.
- It also means that the customer could also be billed for their VIM as opposed to it being part of the shared infrastructure.

VIM on-demand Deployment

- A VIM needs to be allocated for each slice.
- We need a management component in each domain which can allocate a slice.
- A slice owner can manage, configure, and control their own VIM.
- The actual VIM that is deployed could be one of many:
 - e.g. OpenStack, OpenNebula, OpenVIM
- The VIM will be chosen from a pre-determined catalogue of VIMs that the DC owner has.
- As a VIM can be its own distributed system we need to decide how and where to deploy the VIM components.

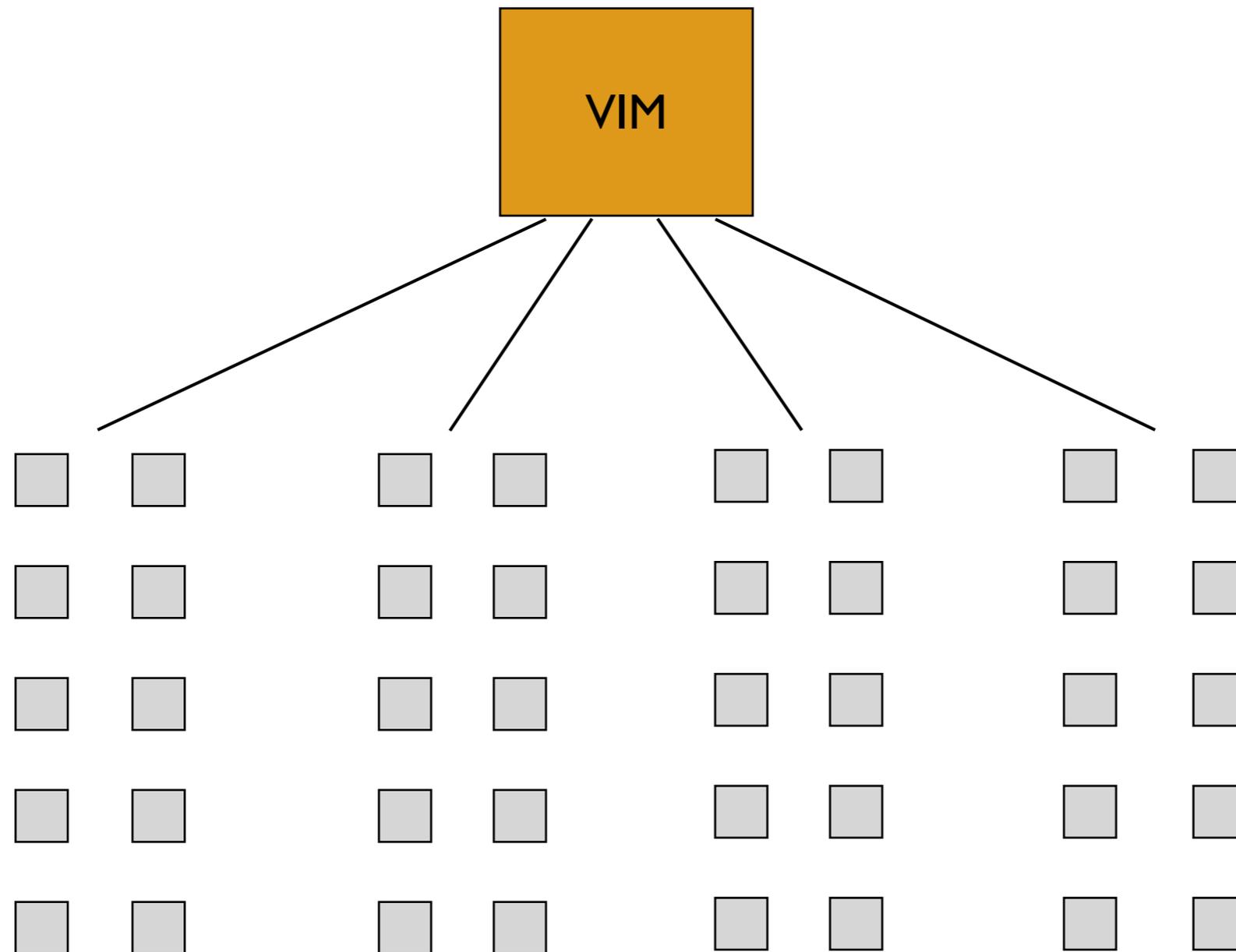
DC Slice Control

- A new component is needed – *the Slice Controller* – to act as a point of control and management.
- The Slice Controller needs to inform each VIM which resources are part of the slice.
- A VIM cannot manage resources that are part of another slice. This is part of the isolation principle.
- The Slice Controller makes these updates on-the-fly as a slice can grow or shrink at runtime.
- Slice Access Control has to be part of the Slice management functionality.

Structural View

- The following slides show a structural view of how a DC can be sliced.
- We see it from the viewpoint of a single Data Center.

One VIM per DC

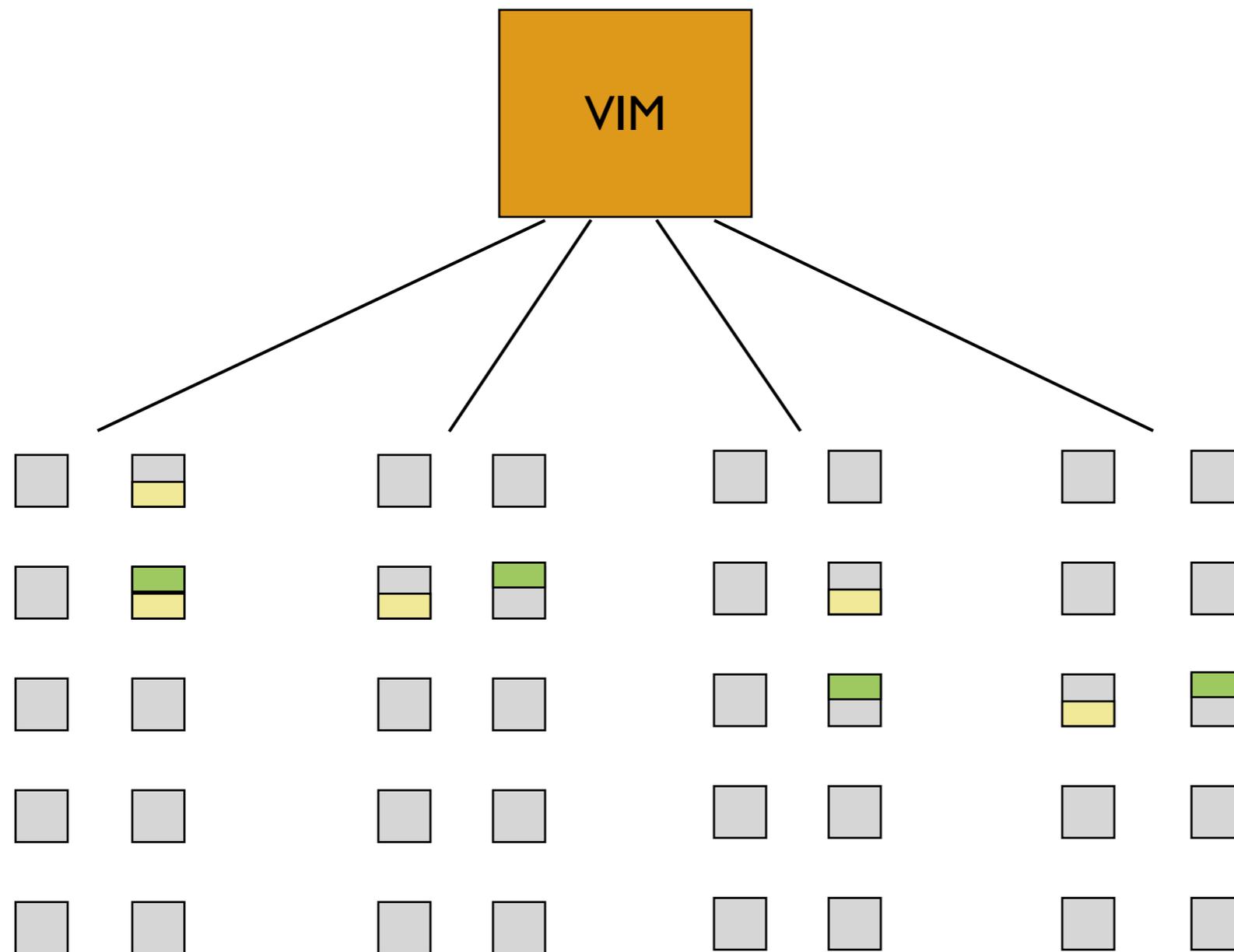


This approach has one VIM for all of the resources of the DC.

This is the current situation in DCs.

The VIM gets more and more functionality in order to do more or different tasks.

One VIM per DC



This approach has one VIM for all of the resources of the DC.

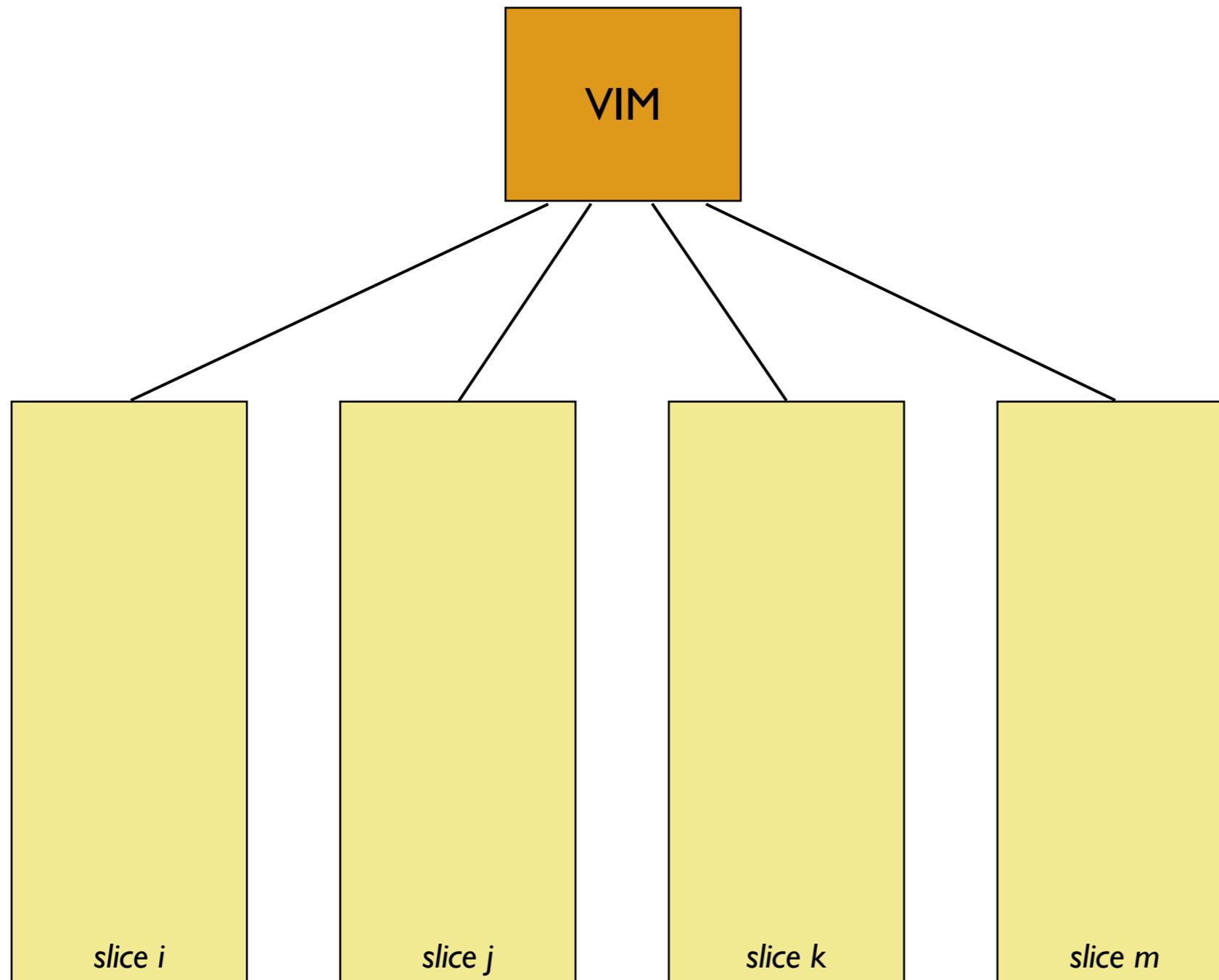
This is the current situation in DCs.

The VIM gets more and more functionality in order to do more or different tasks.

The VMs for the NFVs are scattered across the infrastructure, and each slice is intermingled with the others.

The attributes of the network part of the slice do not apply in the DC.

One VIM for all slices



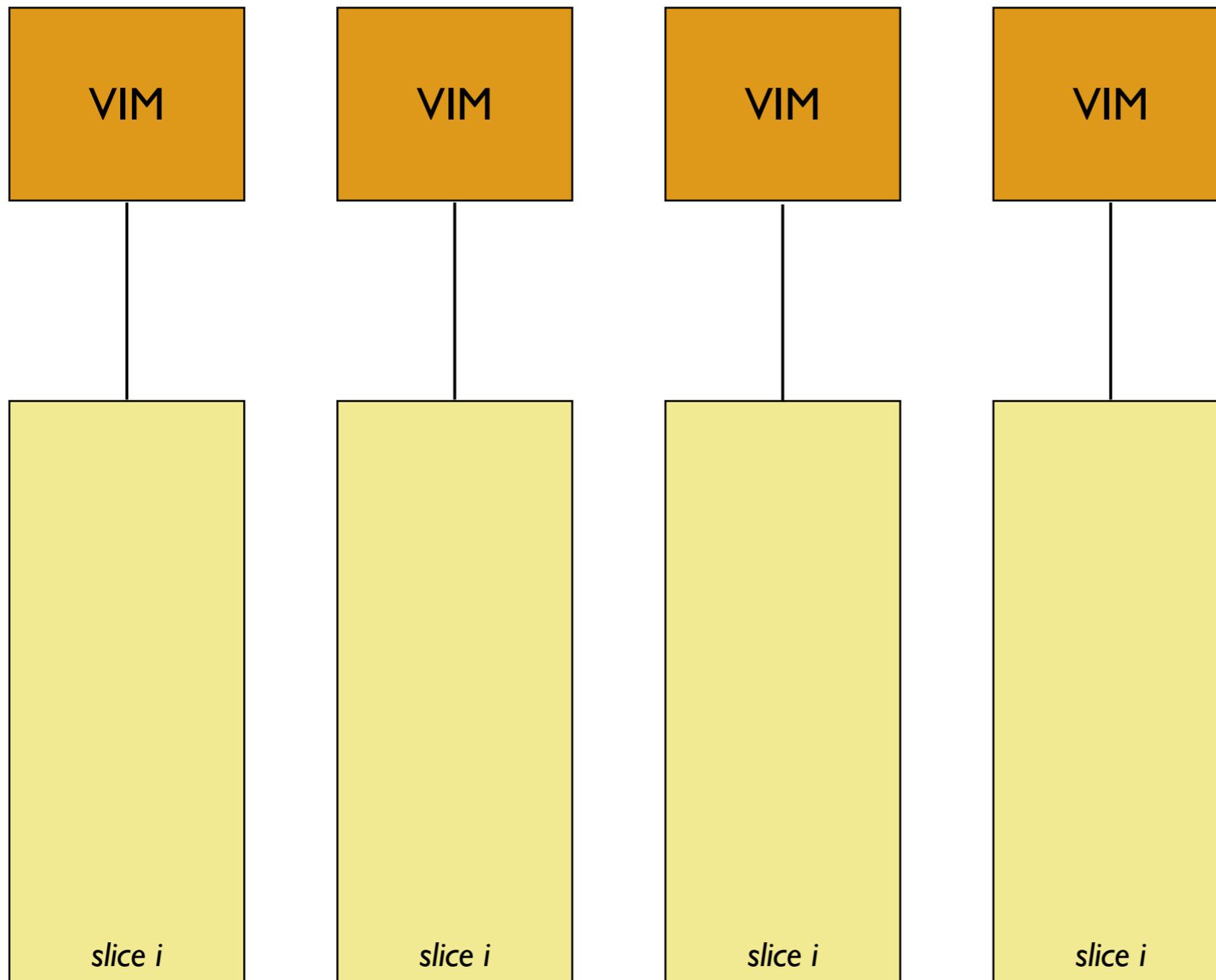
The approach of one VIM for all of the resources of the DC has many issues when we introduce slices.

It forces all of the slices to have the same strategies and policies, as there is only one VIM.

The owner of each slice does not have the flexibility to control the slice how they wish, with different placement strategies of different approaches to energy management.

The VIM needs to be even more complex to deal with this.

One VIM per Slice



This approach has many VIMs - one per slice.

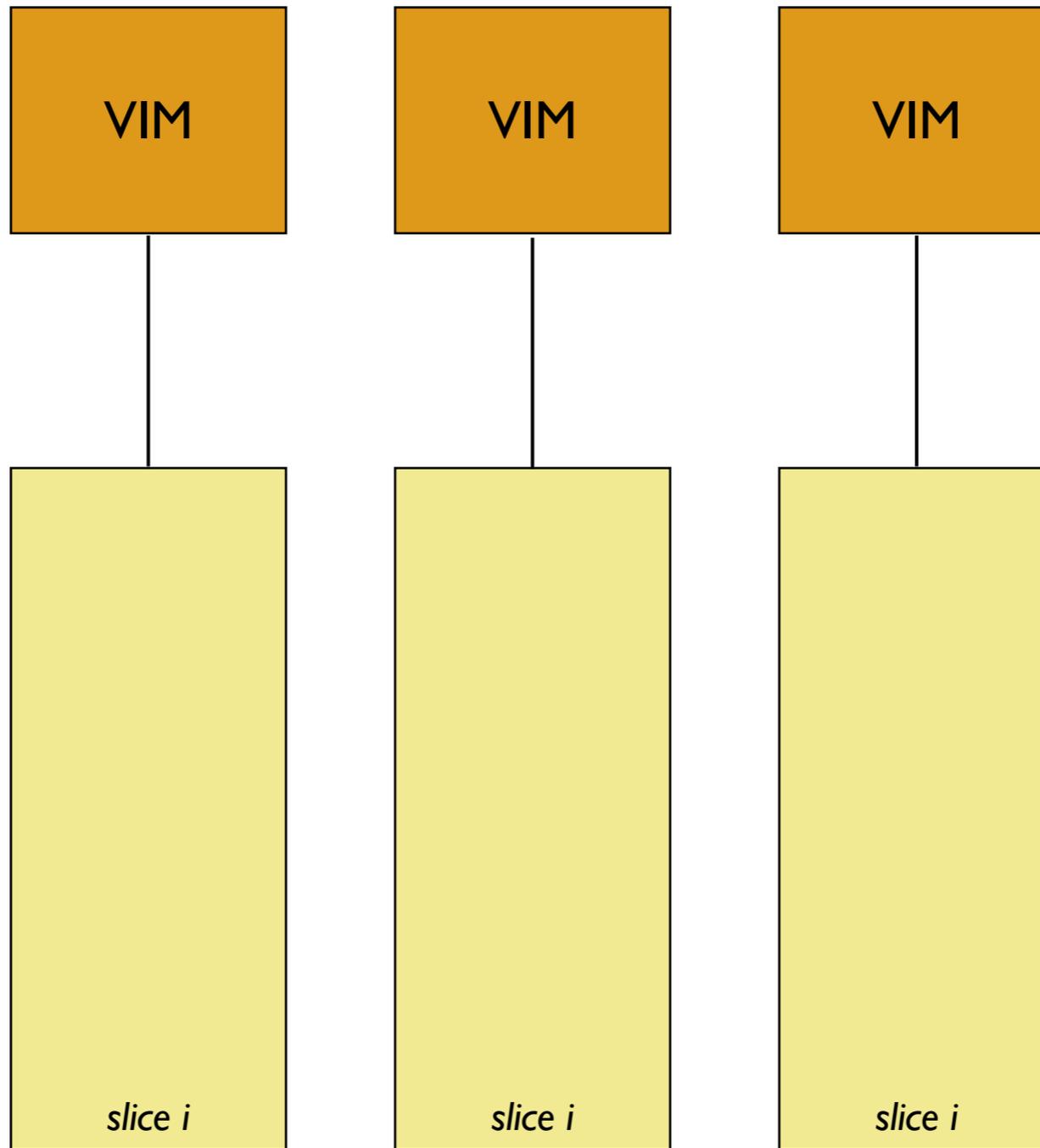
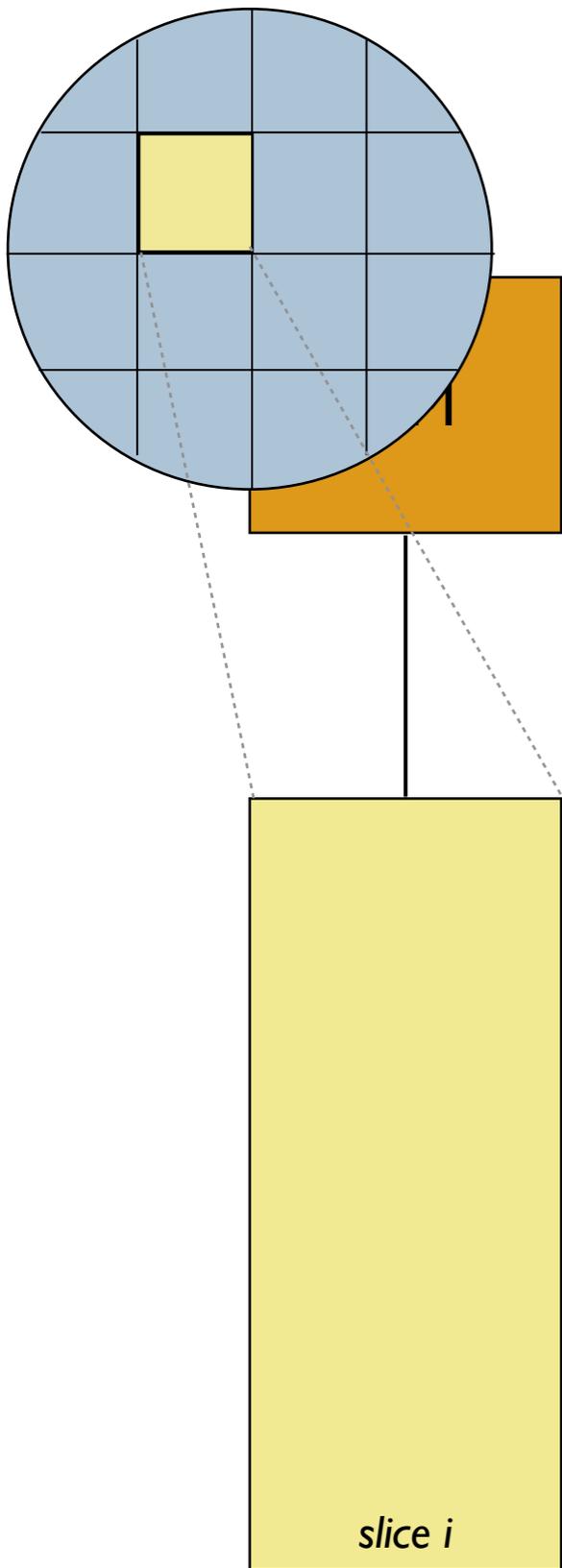
Each VIM can have its own independent strategies and can be managed differently from the other slices.

The slice owner can configure the own VIM as needed for their use.

We need to ensure that each slice is isolated from the others, and that a VIM cannot manage the wrong resources.

The one VIM per slice is a basic premise of our approach.

One VIM per Slice



This approach has many VIMs - one per slice.

Each VIM can have its own independent strategies and can be managed differently from the other slices.

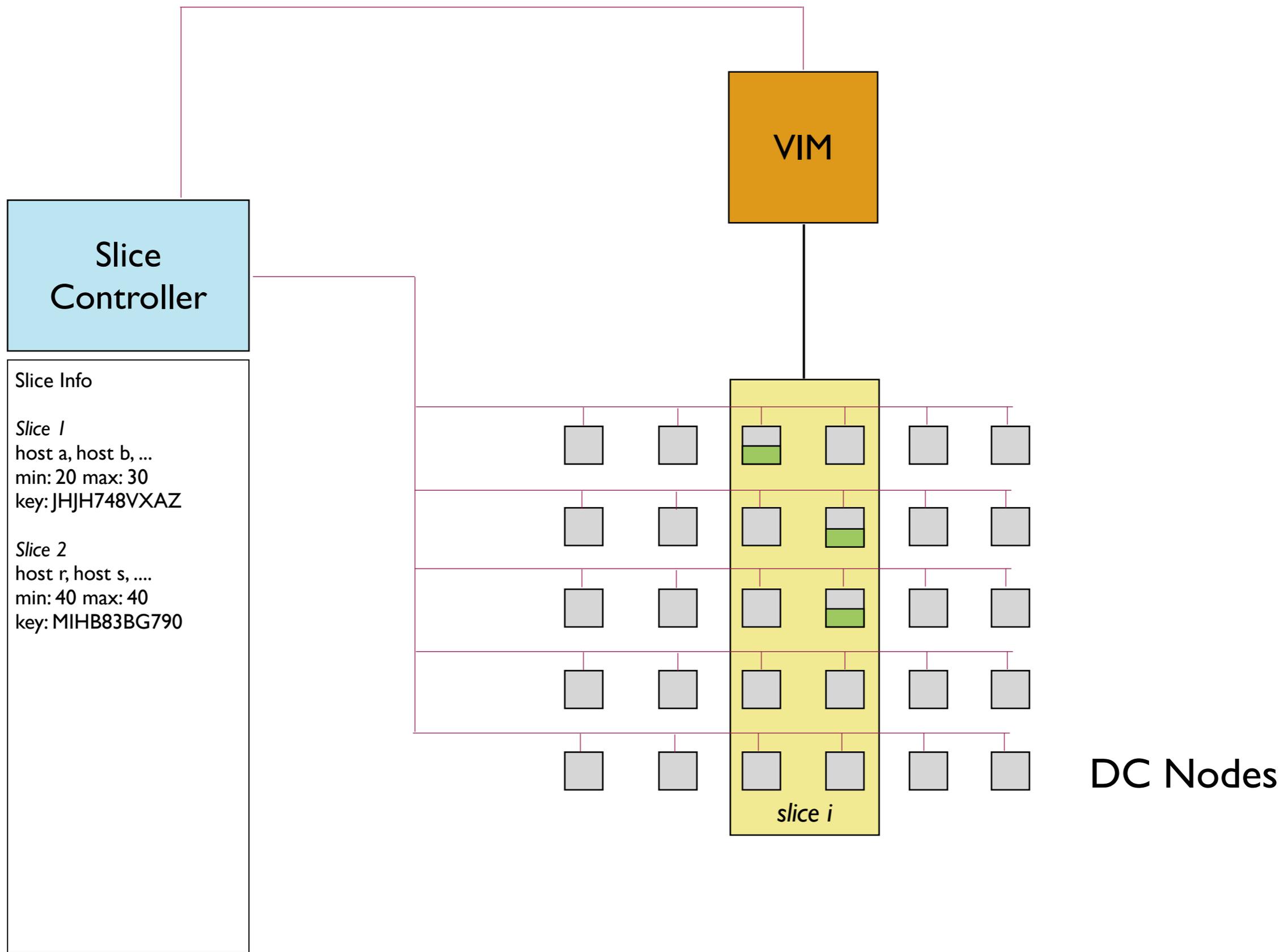
The slice owner can configure the own VIM as needed for their use.

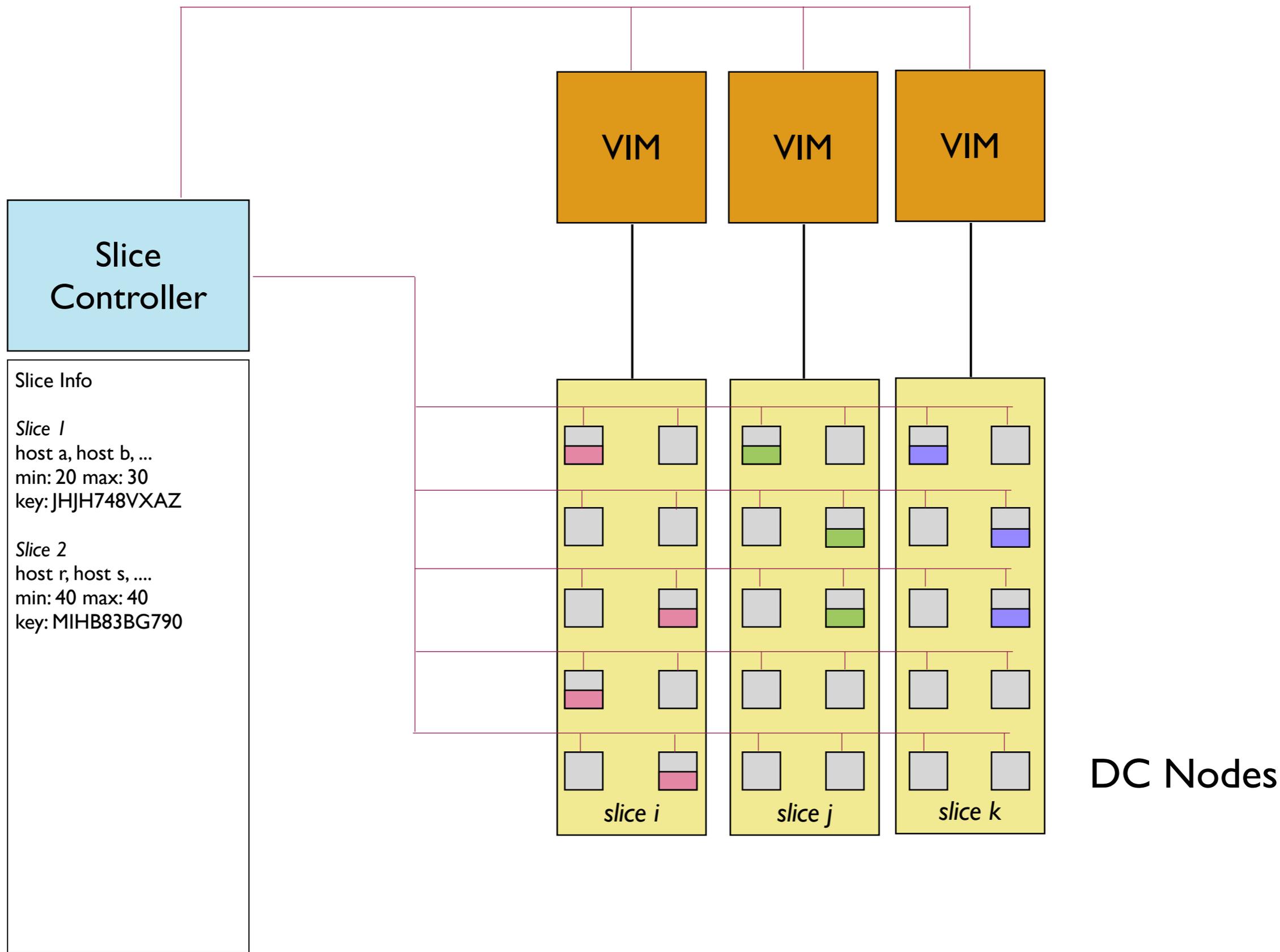
We need to ensure that each slice is isolated from the others, and that a VIM cannot manage the wrong resources.

The one VIM per slice is a basic premise of our approach.

UCL Proof of Concept

- To support service provisioning, we have built a mechanism to support the slicing of the resources.
- To manifest this slice approach, we have designed a built a Slice Controller which is able to allocate a slice of a network data centre (DC) and create a per-slice VIM in an on-demand fashion.
- The DC slice and the VIM are provisioned solely for use with the service. Each slice and its associated VIM are independent of the other slices and VIMs.
- This is lightweight implementation which can be used at micros DC edge cloud up to big DC. Scalability is a feature of the design.





Overall view

The connected slice elements are aggregated into a topology, but for each slice there needs to be a mechanism to interact with it. Also the slice can come from a different provider.



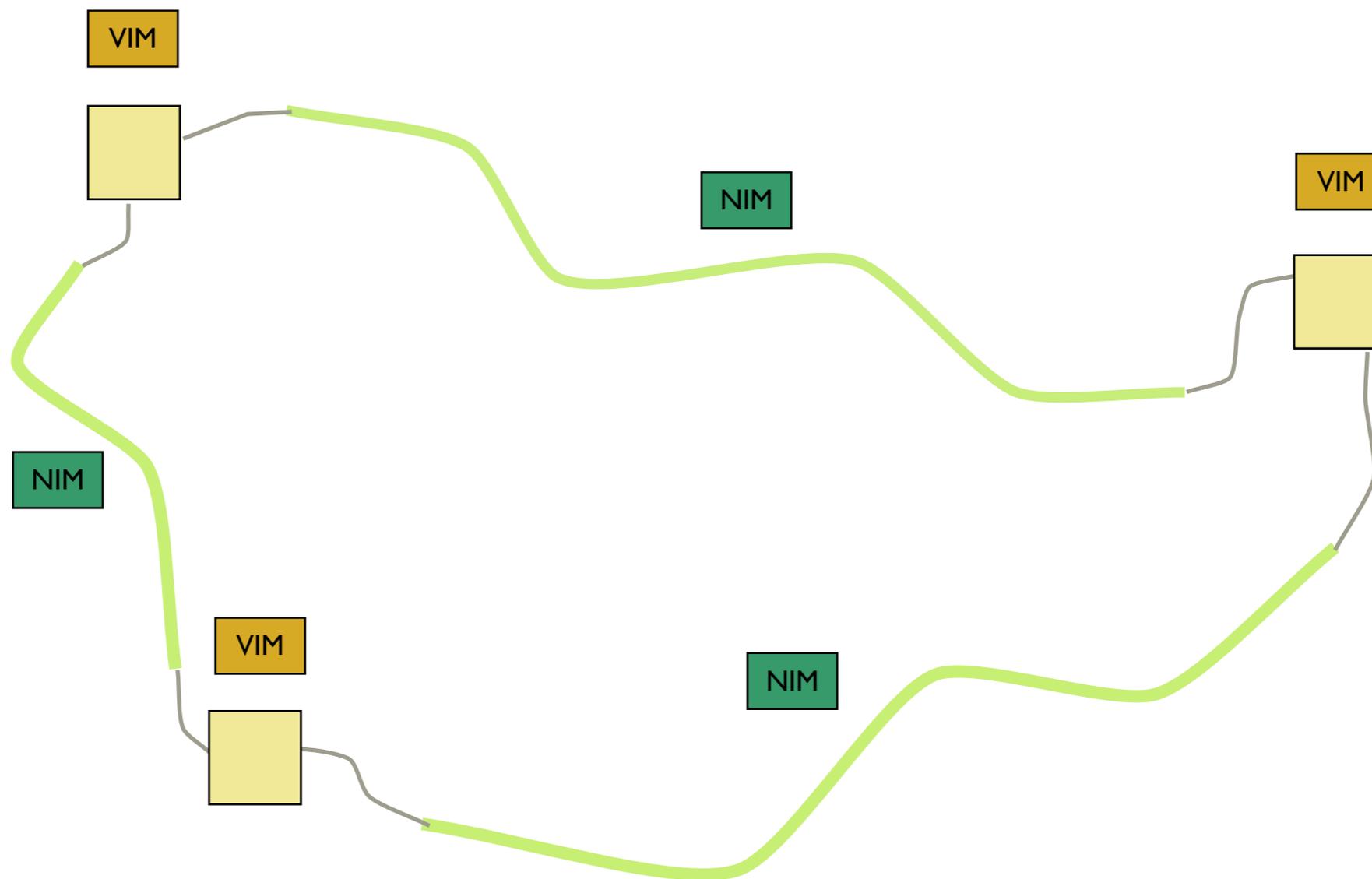
Overall view

Each slice element should have its own manager. For the DC slices we can use a VIM, as this is already known to work well for servers in DCs.



Overall view

For networks we can and should have a similar mechanism, called a NIM (maybe) which acts as an equivalent management point of interaction and control.

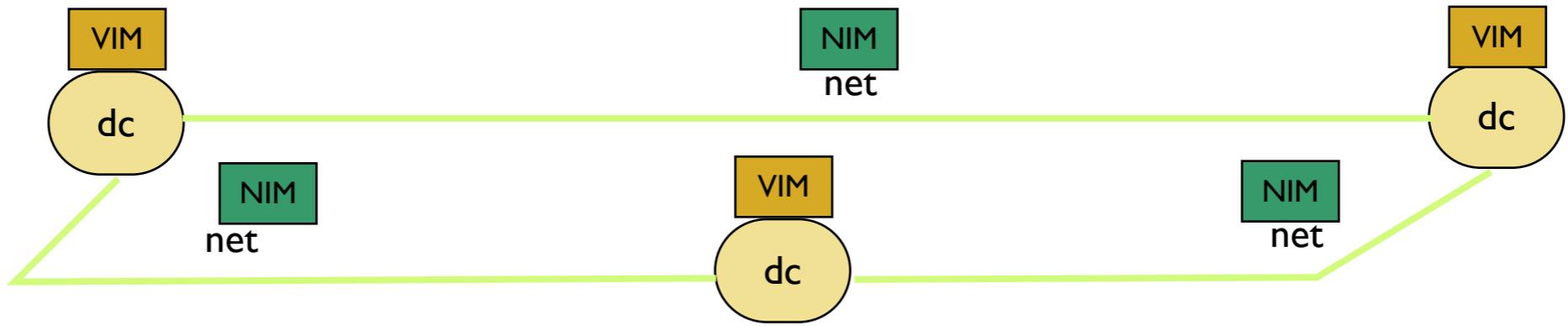


Orchestration



Has representation of the slice as an abstraction

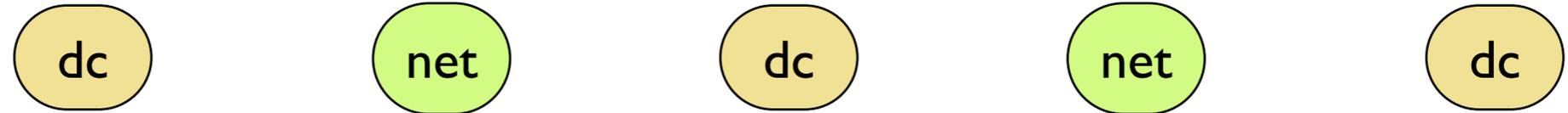
Slice Topology



Slice Management



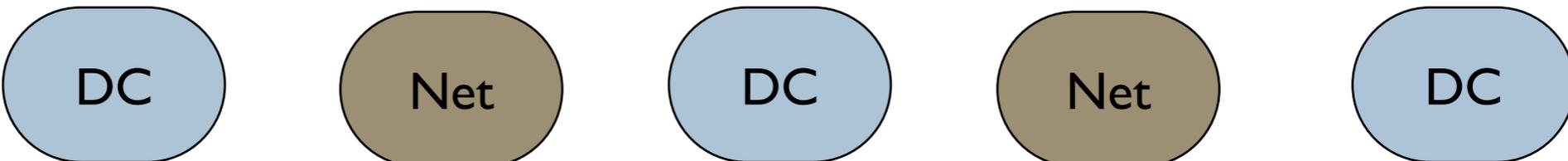
Slice / Partition



Slice Control



Resource

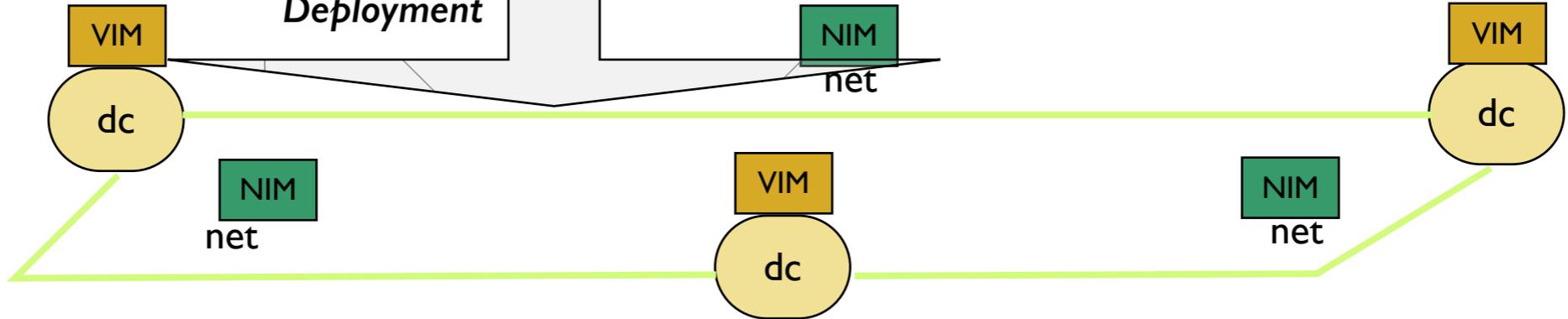


Orchestration



*Service
Deployment*

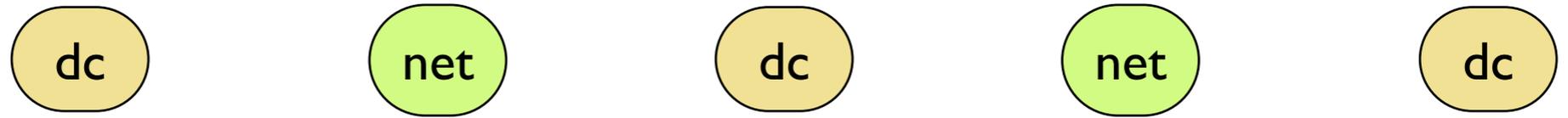
Slice
Topology



Slice
Management



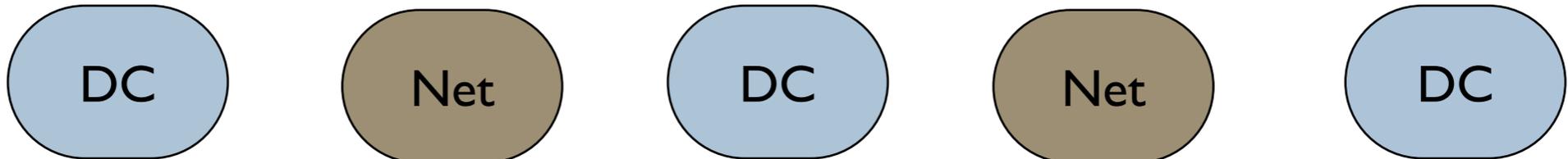
Slice /
Partition



Slice
Control



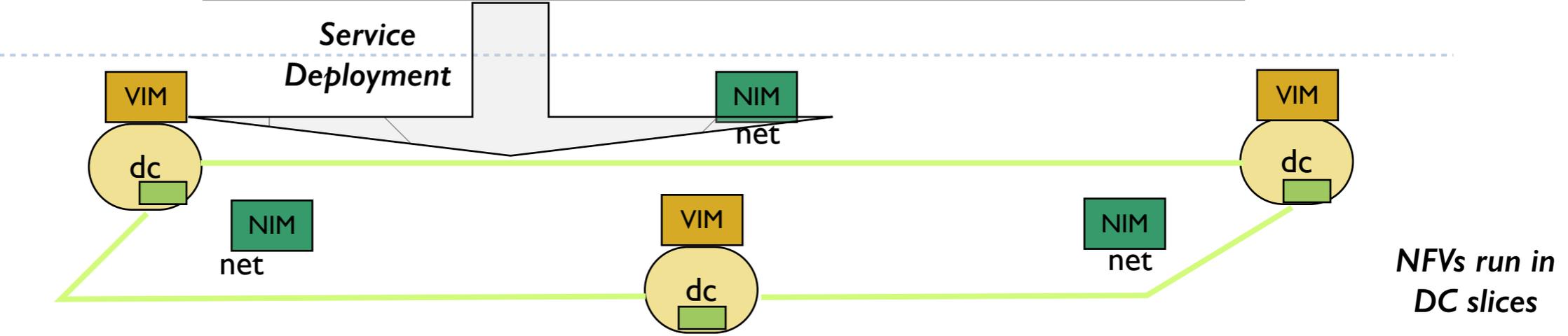
Resource



Orchestration



Slice Topology



Slice Management



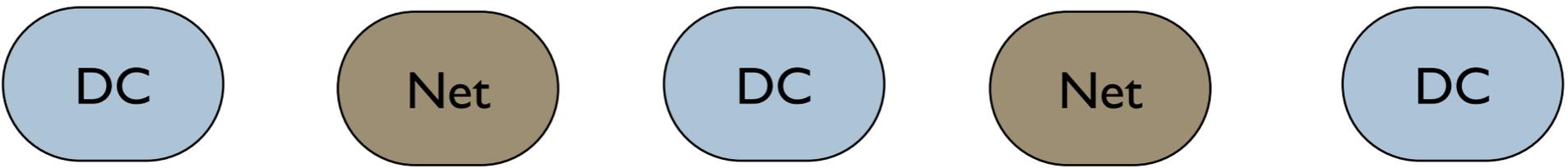
Slice / Partition



Slice Control



Resource



Conclusions

- There are some scenarios in which it is important to have a separate Data Center slice within a full Network Slice.
- We presented the case for creating a VIM on-demand for these DC slices, show some of the attributes of such a slice, and the increased value of a per slice VIM.
- The architectural elements that are required to support such a model are shown.
- Finally a set of layered abstractions using slicing elements were presented, showing how they all fit together for service provisioning.

Conclusions

- These are all part of a bigger picture for control and orchestration.
- Such an approach is highly suitable to the use of clouds in the context of networks.
- We see that there is a link between on-demand VIMs and NFV deployment capabilities.
- For real-life deployment, this approach needs to be linked with functionality and APIs of an orchestrator.
- Further work is in the area of standards, API specification, slice models, interoperability, etc.

Acknowledgements

▶ UCL

- Dario Valocchi, Francesco Tusa, Alex Galis

▶ 5G PPP EU Research Projects:

- 5GEx – “5G Multi-Domain Exchange” <https://5g-ppp.eu/5gex/>
- 5G SONATA – “Service Programming and Orchestration for Virtualized Software Networks in 5G” <https://5gppp.eu/sonata/>.
- NECOS – newly funded project on Network Slicing

▶ Downloads from <http://clayfour.ee.ucl.ac.uk/>