Federation-less Federation of ProtoGENI and VNode Platforms

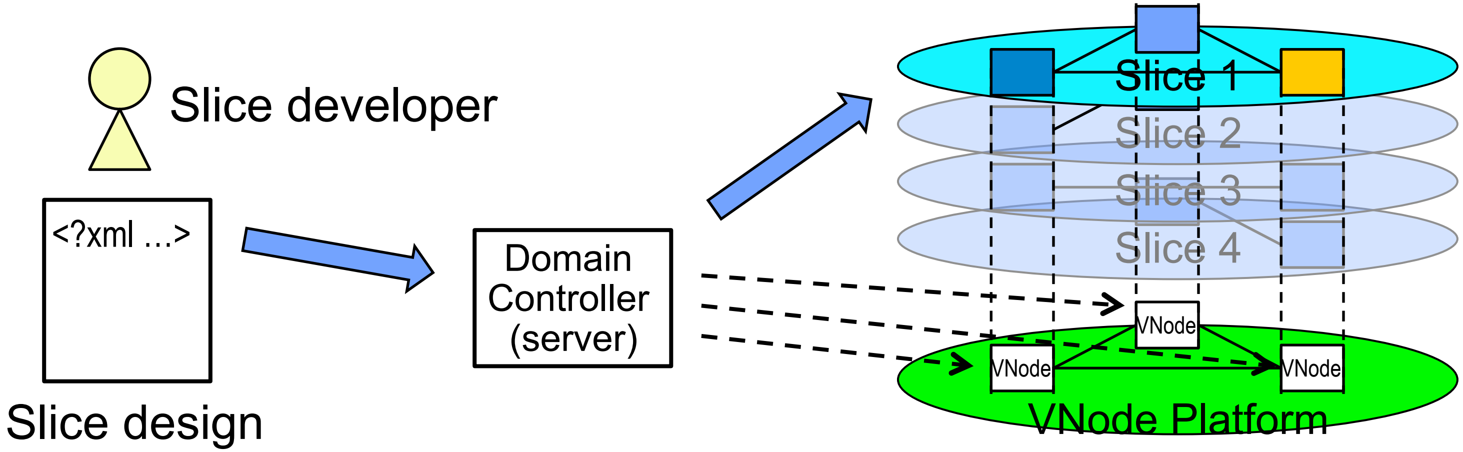
Yasusi Kanada and Toshiaki Tarui
Hitachi, Ltd.

Outline

- ► **VNodes and VNode Platforms that create and manage virtual networks have been developed.**
- ► **We have developed a method for federating virtualization platforms with two features.**
  - ■ **Federation-less federation**
    - • enables federation of VNode platforms that do *not* support federation functions.
  - ■ **Familiar single-interface federation (FSIF)**
    - • allows slice developers to use their familiar single-interface to manage federated slices.
- ► **To apply this method to federation of VNode and ProtoGENI (in US), several problems must be solved.**
  - ■ **Problem**: Although ProtoGENI has federation function, it does not support FSIF, i.e., our federation method.
  - ■ **Solution**: We applied the federation-less federation method to ProtoGENI as well as VNode to enable the federation.

Introduction to VNode and VNode Platform

- ► **We have developed VNode and VNode Platform in a collaborative project.**
- ► **VNode** is a *deeply-programmable* physical node with network-virtualization function.
  - ■ Deeply-programmable: packet data processing, such as new *non-IP* protocol processing, can be programmable.
- ► **Slice** (virtual network) on a VNode Platform is created by sending a slice design to the domain controller (a server).



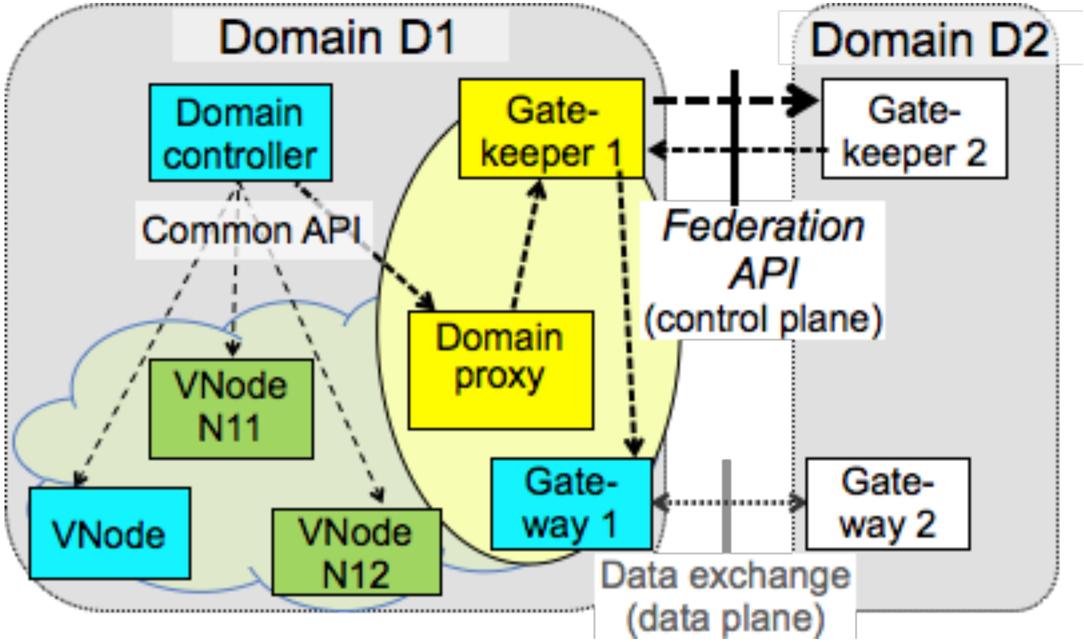
Federation method developed for VNode Platform

- **A method for federating two or more heterogeneous virtualization platforms including VNode Platforms was developed.**
  - This method extends the homogeneous federation method proposed by the authors. (Kanada, et al. 2013)

- **This federation method has two features.**
  - **Federation-less federation**
  - **Familiar single-interface federation (FSIF)**

[Feature 1] Federation-less Federation

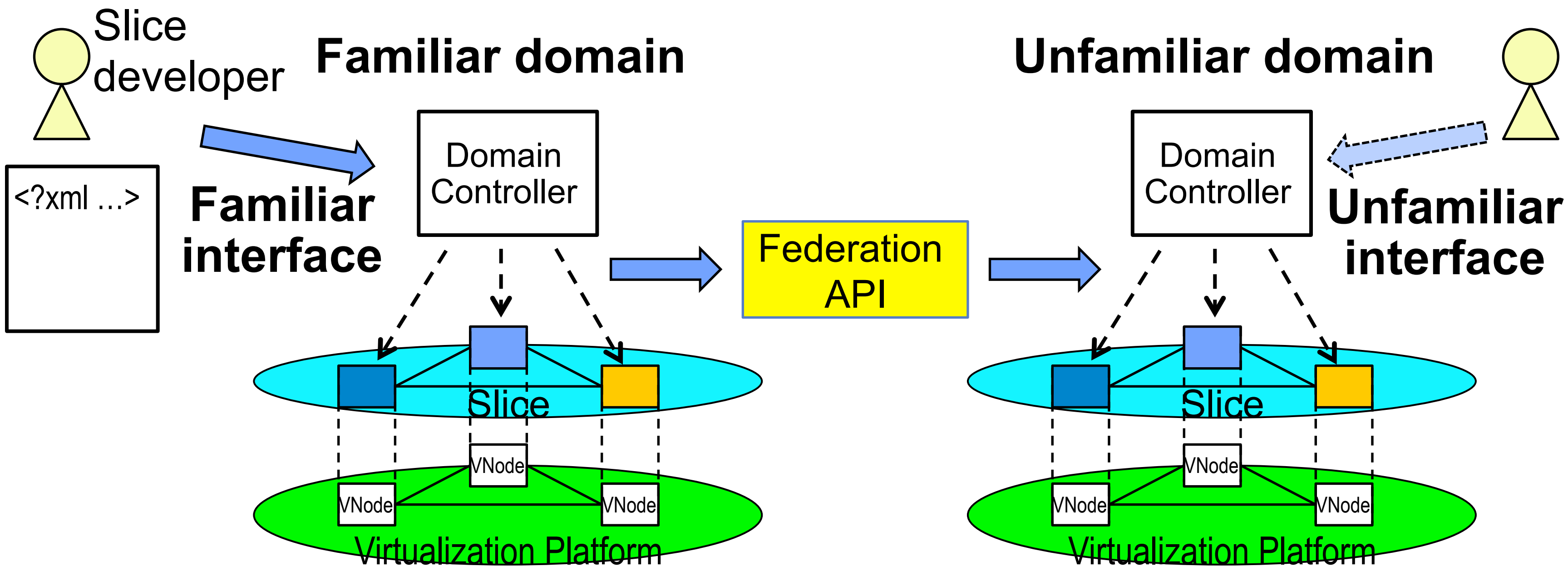
- **This federation method enables federation of multiple domains that do *not* support federation functions.**
  - The original VNode Platform does not have federation function.
  - Adding several components to the original platform and updating the configuration enable federation without modifying the domain infrastructure.



- **We originally developed this method for homogeneous federation of two or more VNode domains.**

[Feature 2] Familiar Single-interface Federation (FSIF)

► **This federation method** (Kanada, et al. 2013) **provides a single management interface to slice developers.**



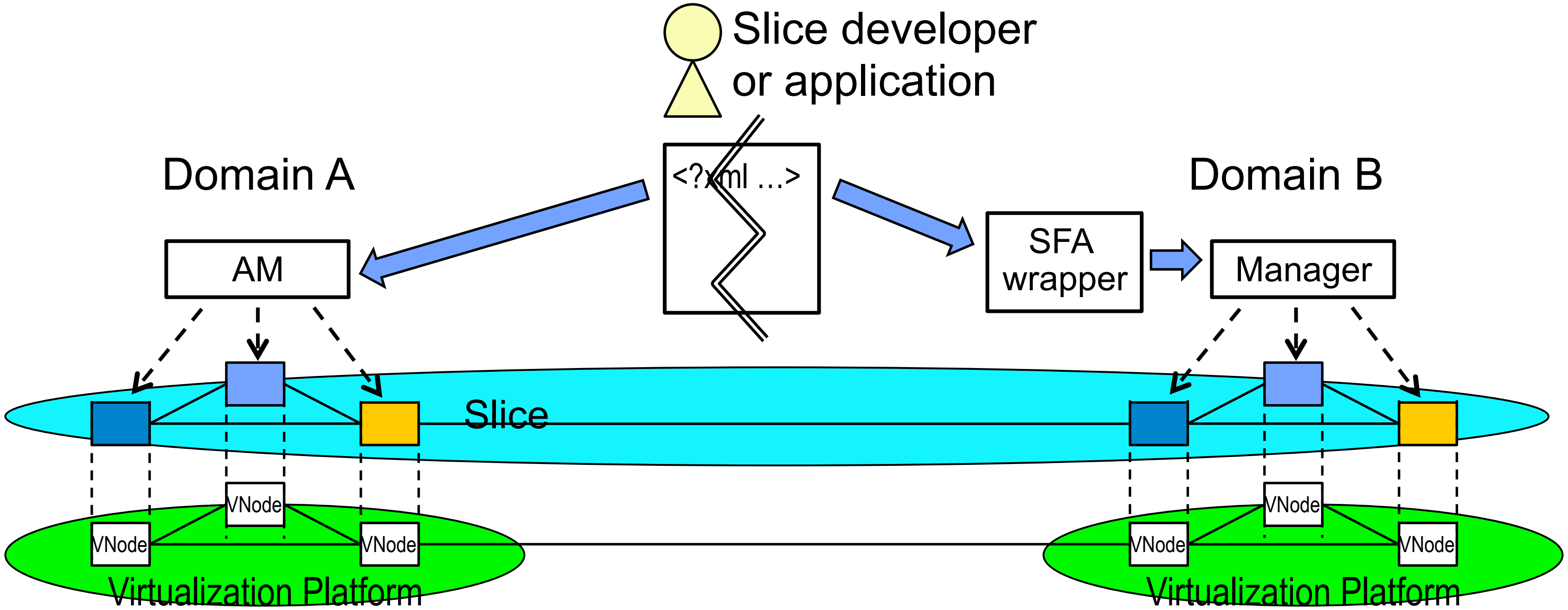
► **A federated slice is created by a negotiation between the domains.**

Related Work: GENI, ProtoGENI and SFA

- ► ***GENI*** (Global Environment for Network Innovations) is a collection of projects and testbeds in the US.
  - ■ GENI “provides a virtual laboratory for networking and distributed systems research and education”.
  - ■ GENI “promotes innovations in network science, security, services and applications”.
- ► ***ProtoGENI*** **is a domain (called an “aggregate”) of GENI.**
- ► **In GENI, aggregates are federated by the *SFA*** (slice-based federation architecture)**.**
- ► **Many other platforms support GENI-style (SFA-based) federation by using *SFA wrappers*.**

Related Work: GENI, ProtoGENI, and SFA (cont'd)

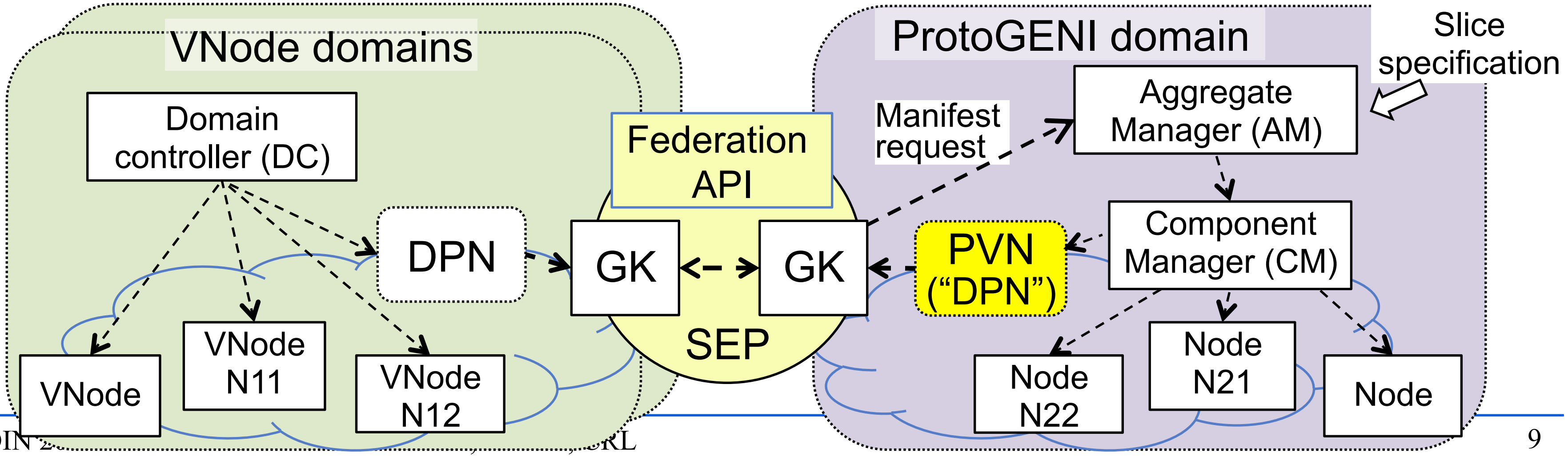
- **GENI does not support single-interface federation.**
  - A slice developer or an application must send slice fragments to all the managers of the aggregates (AMs).



- **This federation method is far from ideal even if the interfaces are unified (to the SFA).**
  - The operation sequence, which is quite different from single-domain sequence, is strongly dependent on the domains to be federated.

Federation-less Federation for ProtoGENI

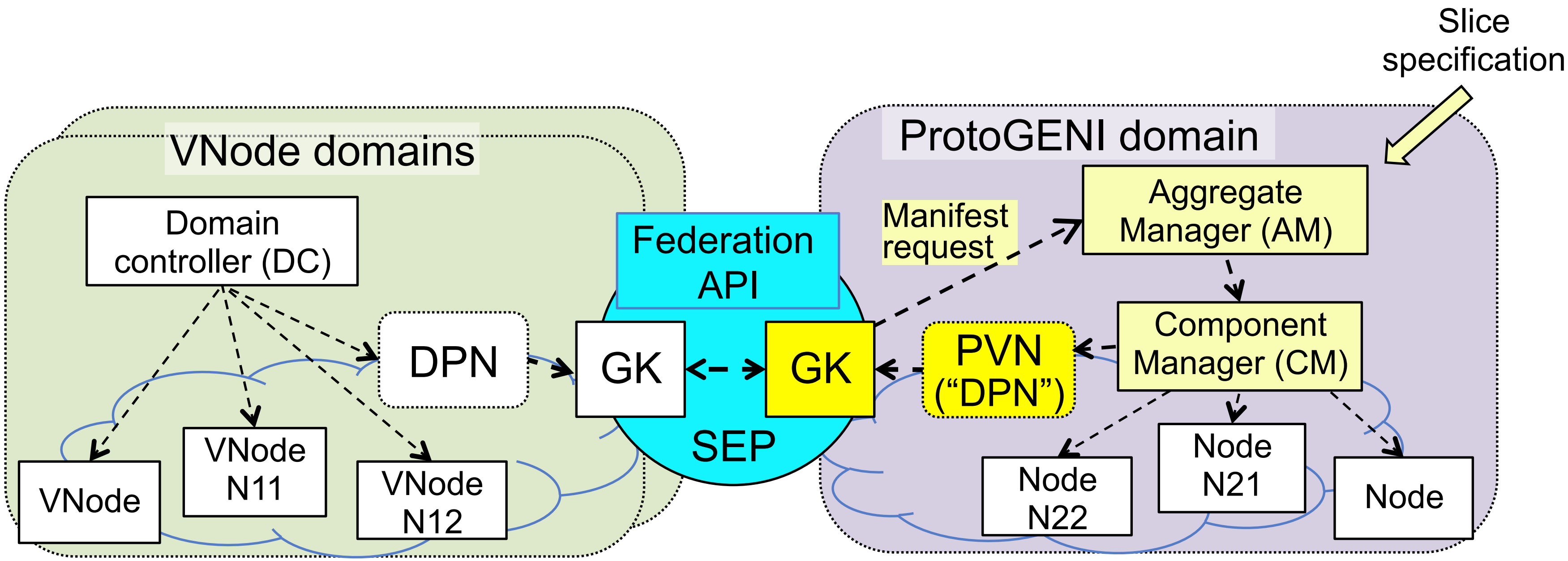
- ► **We intended to federate VNode Platforms and a ProtoGENI aggregate.**
- ► **To apply FSIF, two problems must be solved.**
  - ■ **Problem 1**: The federation method built into ProtoGENI, i.e., SFA, does not support FSIF.
  - ■ **Solution 1**: Instead of using SFA, we applied the federation-less federation method to ProtoGENI, as well as VNode.
  - ■ **Problem 2**: Federation-less federation requires platform component addition and config update, but we could not modify ProtoGENI config.
  - ■ **Solution 2**: A platform component called DPN (domain proxy node) was replace by in-slice component (i.e., PVN, pseudo virtual node).



Federation-less Federation for ProtoGENI (cont’d)

► Method to apply FSIF to ProtoGENI

- The messages between the domains are treated by the federation mechanism (i.e., a SEP and two GKs).
- When the “DPN” (actually an in-slice component PVN), is created, it invokes the GK (i.e., a physical component for the federation).
- The GK sends the specifications of the other domains, which are actually retrieved from the AM by a manifest request.



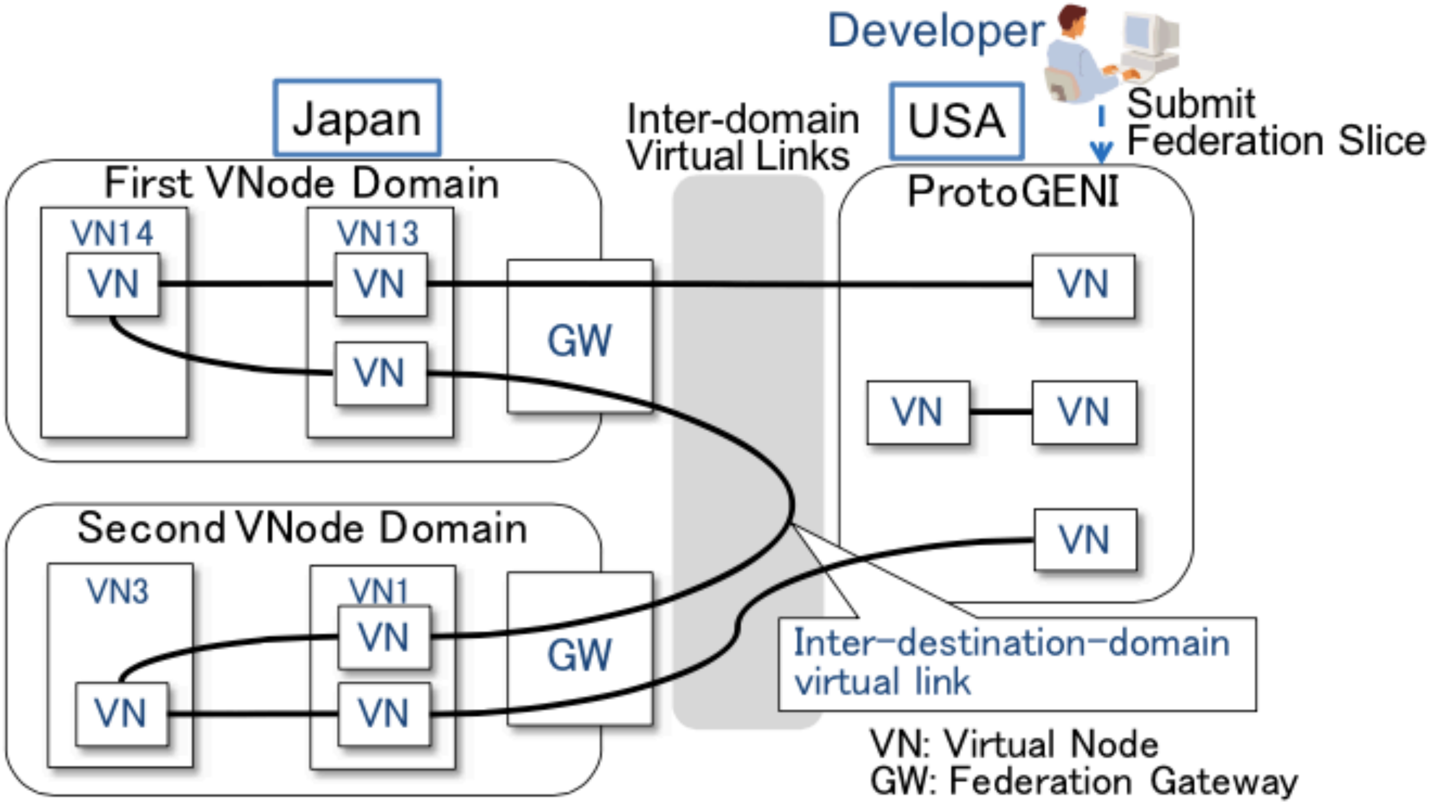
SEP = Slice Exchange Point, GK = GateKeeper

Federation-less Federation for ProtoGENI (cont'd)

- **More problems to be solved**
  - Because a platform component (DPN) is replaced by an in-slice component, more problems must have been solved.
- **[Problem 3] Obtaining RSpecs of other domains**
  - **Problem**: The DPN replacements, at least the GK, need the RSpecs of the other domain, but the RSpecs are not pushed to them.
  - **Solution**: When the GK is triggered by the PVN, it obtains the "manifest" of the slice by sending a "list resources" request to the AM.
- **[Problem 4] Obtaining parameters of allocated resources**
  - **Problem**: The DPN replacements do not receive the parameters for specifying inter-domain connections.
  - **Solution**: Extracting the required parameters, such as the VLAN IDs, from the manifest.
- **[Problem 5] Determining resource allocation/de-allocation timing**
  - **Problem**: Nodes in a created slice do not receive the creation and deletion messages.
  - **Solution for slice creation**: To create the VNode-domain part of the slice when the ProtoGENI-domain part is started.
  - **Solution for slice deletion**: To send a deletion request, which is called "delete-slivers" in GENI, to the client proxy or the PVN instead of sending it to the AM.

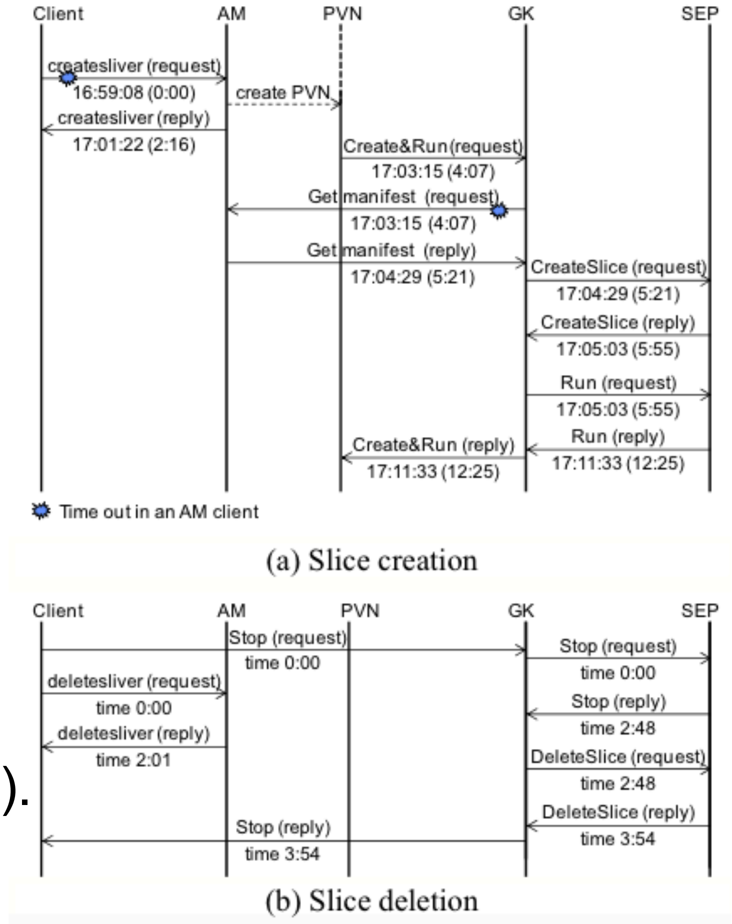
Implementation and Evaluation

- **The federation functions were partially implemented for demonstration and evaluation.**
  - A slice specification was distributed between a ProtoGENI aggregate in the University of Utah and two VNode domains in Japan.
  - The domains were connected through trans-pacific VLANs.



Implementation and Evaluation (cont'd)

- ► **Messaging for creation and deletion of the federated slice was captured.**
  - ■ The results shows that the time required for slice creation is acceptable, but still much longer than the optimum.
  - ■ The results shows that the time required for slice deletion is reasonable, but the sequence should be improved in future.
- ► **The implemented federation method was successfully used for a demonstration.**
  - ■ Federation of ProtoGENI and VNode domains was demonstrated in GEC 20 (the 20th GENI Engineering Conference).



(a) Slice creation

(b) Slice deletion

Conclusion and Future Work

► Conclusion

- The proposed method was successfully used for demonstrating federation of the ProtoGENI and two VNode domains.
- This method was proved to be useful for experiments and demonstrations.

► Future work

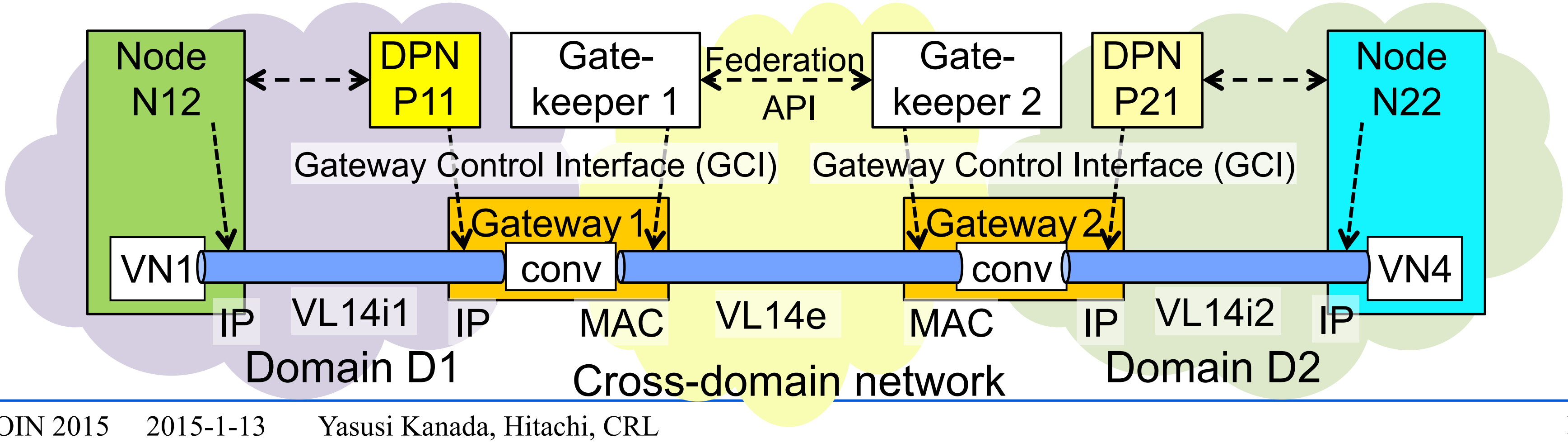
- To improve the implementation of the proposed method for the ProtoGENI platform.
- To improve the slice-deletion sequence.
- To apply this method to federation of three or more different types of domains (including a VNode domain).

[Problem 1] Obtaining RSpecs of other domains

- **Problem: The GK (and PVN) need the RSpecs of the other domain, but the RSpecs are not pushed to PVN.**
  - In the original design, a DPN receives the whole slice specification that contains a specification of a PVN, which contains the RSpecs of the other domain.
  - However, because the PVN is a in-slice component, it does not receive the RSpecs.
- **Solution**
  - When the GK is triggered by the PVN, it obtains the “manifest” of the slice by sending a “list resources” request to the AM.
    - A manifest is an XML-based description that contains both the RSpecs and parameters assigned as the results of resource allocation; that is, it contains both top-down and bottom-up information concerning the slice.
  - The GK extracts the RSpecs of the other domains and sends them to the domains through the SEP.
    - Because the manifest contains a copy of the specification of the PVN that contains the original RSpecs of the virtual nodes and links, which are described by the slice developer, it contains the RSpecs of the other domains.

[Problem 2] Obtaining parameters of allocated resources

- **Problem: The DPN replacements do not receive the parameters for specifying inter-domain connections.**
  - In the original design, domain-internal parameters for connecting inter-domain links are obtained by negotiation; however, no such mechanism is available on ProtoGENI.
  - Each inter-domain link is divided into three parts, i.e., an inter-domain part and two intra-domain parts.
  - A DPN in each domain receives the parameters for the intra-domain part by negotiating with the end-point VNode; however, in a ProtoGENI domain, no negotiation is used for this purpose.
- **Solution: Extracting the required parameters, such as the VLAN IDs, from the manifest.**
  - The PVN sends these parameters to the other domain through the SEP.



[Problem 3] Determining resource allocation/de-allocation timing

► Problem: Nodes in a created slice do not receive the creation and deletion messages.

- They are merely created when the slice is “started”, and it is merely killed when the slice is “stopped”.
- The DPN replacements are thus unable to catch the timing of slice creation or deletion.

► Solution for slice creation

- To create the VNode-domain part of the slice when the ProtoGENI-domain part is started.
- This solution is acceptable when the size of the slice is not large.
- However, if a more scalable solution is required, an alternative solution can be applied (but omitted here).

► Solution for slice deletion

- To send a deletion request, which is called “delete-slivers” in GENI, to the client proxy or the PVN instead of sending it to the AM.
- The client proxy or the PVN forwards the message to the AM and the GK.
  - If the PVN is used for this purpose, this messaging must precede destruction of the PVN.
  - If the client sends the message to the GK or the client proxy, this constraint is not required, and this method is symmetric; however, the client must know the address and interface of the GK.