GENI Network Stitching - Overview

Table of Contents

GENI Network Stitching Architecture -Overview

1.0 Introduction

This purpose of this document is to present considerations and options for network stitching in the context of the GENI infrastructure. The hope is that this will contribute to a larger set of network stitching objectives which are:

- Initiate a broader discussion within the GENI community regarding network stitching considerations and options for the GENI Infrastructure.
- Provide a framework for more detailed discussion at a Network Stitching Workshop held during GEC10.
- Development of a consensus within the GENI community on a detailed GENI Network Stitching Architecture and Design
- Development of detailed specifications for GENI Network Stitching Components, Protocols, Schemas, and Best Practices.
- Implementation and deployment of a prototype GENI Network Stitching capability

The focus of this document is on the mechanics of network stitching. This implies a focus on the information, intelligence, algorithms, and GENI Substrate/Aggregate Manage capabilities that are needed to realize multi-substrate network stitching. This is generically referred to as network stitching mechanics in this document.

There is the associated topic of network stitching related user and resource authentication, authorization, and policy application. This is generically referred to as network stitching policy in this document. While network stitching policy is a very important set of features and capabilities, this document will focus on the network stitching mechanics for the following reasons:

- treating these two topics as separate but related will maximize the ability to make progress in both areas
- the resources associated with network stitching can be viewed as just another set of resources to be allocated or federated via integration the GENI-wide Slice Federation Architecture (SFA). As a result, once the GENI network stitching mechanics are agreed upon, the SFA user credential and slice (resource) credential structure already developed within GENI can be applied to the network stitching components.
- due to the work on the GENI SFA, user and resource authentication/authorization have been much more widely discussed within the GENI community as compared to the issues associated with network stitching mechanics.

For these reasons, the remainder of this document will focus on issues associated with network stitching mechanics.

2.0 Network Stitching in the GENI Context

In the GENI context, network stitching primarily refers to the interconnect of multiple substrates, each of which are under the control of their Aggregate Manager. The interconnect here is at the network level and typically refers to an Ethernet VLAN based stitching function. However, other methods and technologies could be utilized and the network stitching architecture and design should accommodate those as well. The other technology options include Wavelength Division Multiplexing (WDM), Synchronous Optical Networking (SONET)/Synchronous Digital Hierarchy (SDH), IP Routing, IP Tunneling, native Optical Carrier Transport, Carrier Ethernet, InfiniBand, wireless link based, heterogeneous combinations of these with horizontal adaptations, various combinations of tunneling of one protocol inside another, and others.

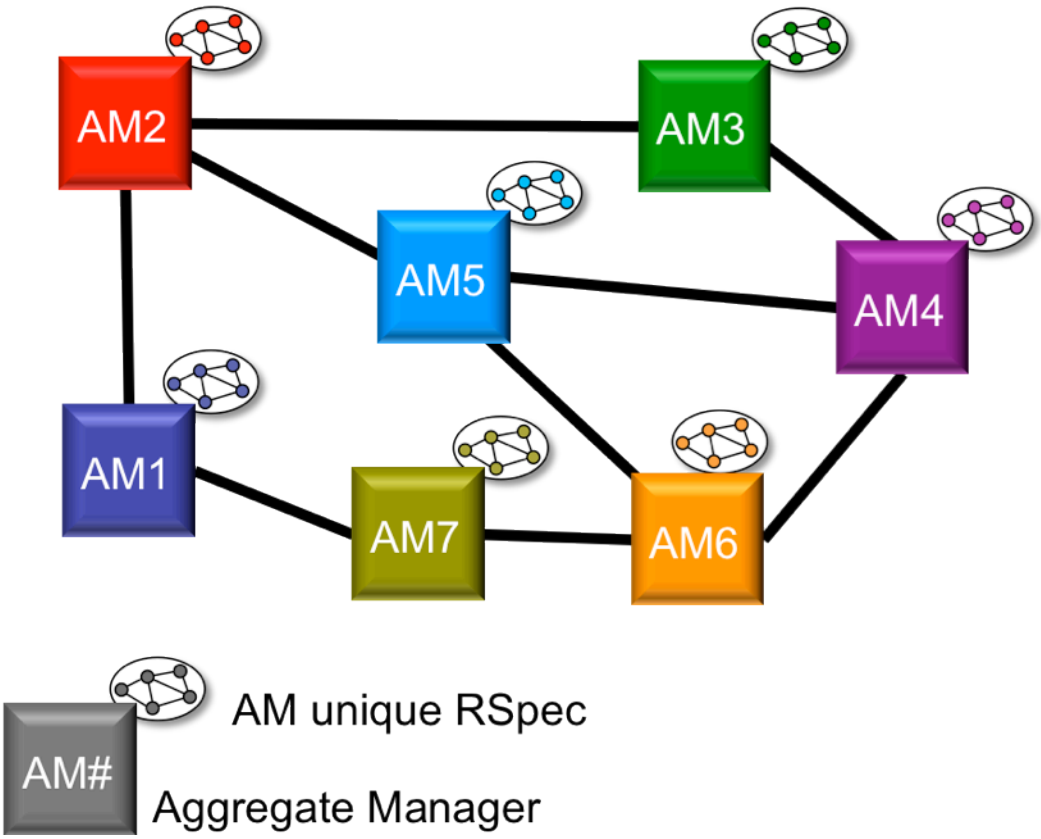


Figure 1. GENI Aggregates - A topology of Substrates

The result of a GENI network stitching operation is really the construction of a "topology" of substrates as represented by their Aggregate Managers in Figure 1. Each Aggregate Manager has a unique RSpec which defines its Substrate resources. These RSpecs are unique in structure and content. However, from a network stitching perspective, we can think of these RSpecs as a topology description of the individual substrate. The result is that we have a topology of substrates, each of which have their own uniquely defined topology (as described by a RSpec).

In actual deployments, GENI substrates will not often be directly connected. Typically there will be other network infrastructures in between GENI substrates. GENI network stitching in this environment will require the static and/or dynamic provisioning of network resources across these external networks. Figure 2 depicts this type of topology. The challenge for the GENI Network Stitching function is to be able to discover the possible options for construction of this substrate topology, identification of stitching-specific resources, and provision of the identified resources.

The GENI Substrate nomenclature refers to the set of physical resources associated with an Aggregate Manager. A specific substrate may be very heterogeneous in terms of types and capabilities of the available resources. It is expected that this substrate topology and capability set will be captured in its unique RSpec. The associated GENI Aggregate Manager (AM) will be responsible for management and control of these substrate resources. The protocols and methods for interacting with an AM is defined in the GENI AM API specification:

- http://groups.geni.net/geni/wiki/GeniAggregateManagerApiDoc

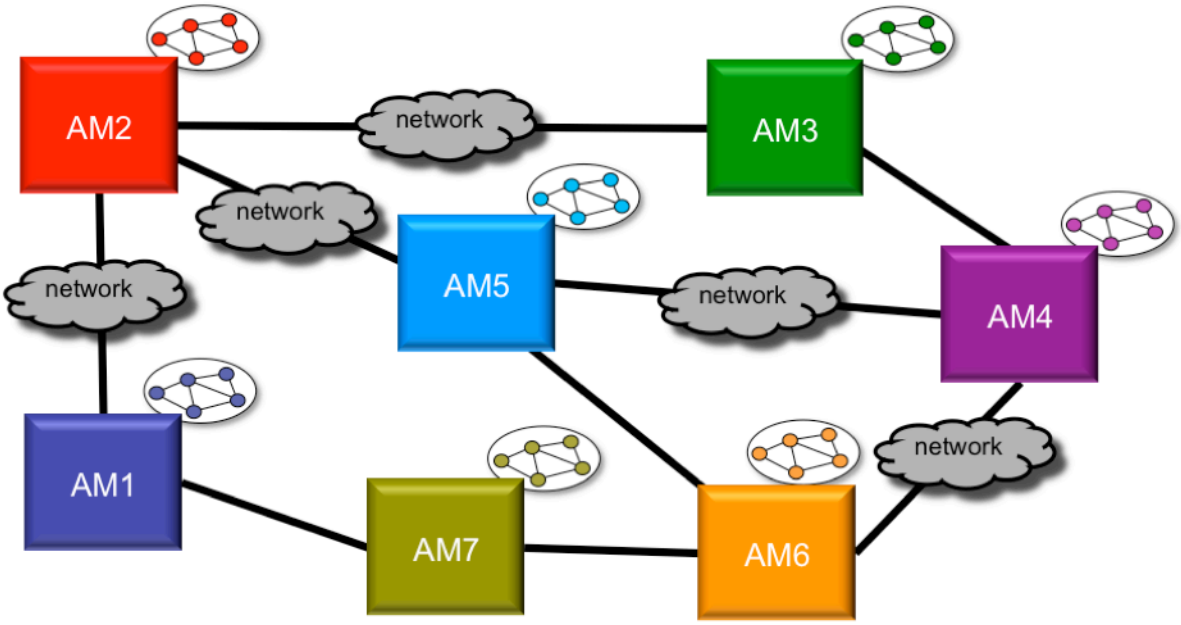


Figure 2. Real World Considerations - External Networks Provide Interconnects

Figure 1 and 2 use the term Aggregate and Aggregate Manager (AM) to represent both the physical topology (substrate) and the management and control function. The remainder of this document will continue to use these terms in this manner.

3.0 GENI Network Stitching - What is needed

The following key capabilities can be identified for the GENI Network Stitching process:

- Ability to create inter-aggregate network connections
- Accommodate interconnects based on Ethernet VLANs initially
- Accommodate interconnections based on other technologies in the future
- Accommodate the fact that external network resources may need to be statically and/or dynamically provisioned in order to realize inter-aggregate network connections.
- Be able to handle slice specific stitching parameters
- Fit into the framework defined by the GENI Aggregate Manager API and RSpec specifications

The following key functions can be identified in order to realize the above capabilities:

- Ability for a specific AM to understand where its external connection points are located and to represent these in its RSpec
- Ability for a specific AM to make available its view of it external connection points
- Ability for the "stitching function" to develop a global view of the physical interconnections between Aggregates based on the AM provided RSpec information, and a representation of the network topologies between them.
- Ability for the "stitching function" to use this global AM inter-connection view to initiate specific resource provisioning actions on AMs and external network resources.
- Ability to accomplish the above in an environment where there may be multiple resource provisioning actions (slice instantiations which include network stitching) simultaneously.

There are multiple architectures and designs possible to realize these functions and capabilities. Development of a specific GENI solution is expected to be the subject of a larger GENI community discussion. However, to facilitate these discussions, an example architecture is described in the following section.

4.0 GENI Network Stitching - Architecture Description

The GENI network stitching functions needs to integrate and be compatible with the key GENI Architecture components including the GENI AM API and the RSpec. Below is some discussion on these components in terms of network stitching considerations.

GENI AM API

The GENI AM API is defined in the below document:

- http://groups.geni.net/geni/wiki/GeniAggregateManagerApiDoc

This API defines the methods and formats for interacting with an AM to accomplish the following with respect to GENI Slice management and control:

- GetVersion
- ListResources
- CreateSliver
- DeleteSliver
- SliverStatus
- RenewSliver
- Shutdown

This API includes the definition of user and resource credentials. There are currently no AM API services directly related to network stitching.

RSpec

The GENI RSpec is the method used by Aggregate Managers to define their aggregate topology, as well as the state of instantiated resources (slivers and slices). The RSpec formats are specific and unique to each AM. A specific RSpec instance may define the entire set of Aggregate resources (topology) or may define a sliver specific set of resources. There has been little focus on including network stitching specific information inside current RSpecs.

The network stitching architecture presented here has the following main components:

- **Stitching Resource Element** - This is an element which would be provided by all Aggregate Managers. The purpose of this component is to define the external connection points for the local AM.
- **Common Stitching Topology Schema** - In order to conduct operations across multiple AMs, the Stitching Resource Element needs to be based on a common topology schema to be used by all GENI AMs. This does not require any standardization or other changes to the structures of the unique AM RSpecs. The intent here is that any entity which can exercise the ListResources (with "all resources" set) feature of the GENI AM API, will get back an AM unique RSpec. In a similar manner the Stitching Resource Element which is formatted in accordance with a GENI standard network stitching topology schema will be made available.
- **GENI Stitching Topology Service** - This is a function which has the ability to collect the Stitching Resource Elements from all the GENI AMs, and build a global view of the inter-connected GENI AM space. The service provided will be to make this global topology view available to other processes, such as the GENI Stitching Function described below, so that the details of a specific network stitching event can be calculated and coordinated.
- **GENI Stitching Path Computation Function** - This function has the ability to obtain the global network stitching topology view from the Stitching Topology Service, perform slice instantiation specific path computations.

- **GENI Stitching Workflow Function** - This function has the ability to take the path computations results and execute the workflow steps required to accomplish network stitching. This includes interacting with the specific AM APIs and external network resources to instantiate the network stitching provisioning actions.
- **GENI AM API Network Stitching Extensions** - Some extensions will be needed to the GENI AM API to support robust and globally scalable network stitching functions. These extensions will revolve around the concepts of stitching specific resource queries, resource negotiations for multi-aggregate stitching operations, basic stitching interfaces, and status interfaces.

There are multiple possible scenarios and sequences for how the above components could interact to accomplish multi-aggregate network stitching. Figures 3a, 3b, 3c, and 3d show several high-level overviews of how this may be accomplished. The intent is that a single architecture and suite of components listed above will support any of these approaches. Additional information and considerations of the numbered items from Figure 3a is provided below.

Figure 3a Information and Considerations:

-Preliminary Steps (not shown in figures): As noted in the top left corner of the figures, some initial steps are not shown in Figures 3a. These include the User/Client obtaining the RSpec for the AMs of interest and constructing the RSpecs that will be utilized in the CreateSliver requests. This is intended to reflect standard GENI practices in this area. The remainder of the enumerated steps are specifically focused on network stitching functions. While these are shown as separate components from the other GENI components, this is only for clarity purposes. There is no reason why these network stitching functions could not be incorporated into the Clearinghouse, client, Aggregate Manager or any other GENI component.

-Step A-Topology Service Data Retrieval: This is the population of the Topology Service with network stitching related topology data. The intention is that each AM will make available a "Stitching Resource Element" via a ListResources AM API call. This component will describe the external connections of the Aggregate. The format of this component will be in accordance with a standard GENI Stitching Topology Schema. Development of a standard format for description of external connections will allow construction of the global GENI aggregate interconnect topology. The exact information contained in the Stitching Resource Element would need to be defined as part of the detailed topology schema specification. However, past experience indicates that the type of information includes link technology type, identification of link remote side, special capabilities (such as VLAN translation), link bandwidth, and other parameters associated with network resource provisioning. It should also be noted that it is possible for these topology descriptions to be updated as the Aggregate information changes. The frequency of updates and the amount of dynamic data that one tries to maintain in the Topology Service can be adjusted to suit specific operational considerations. However, there is generally very little dynamic information maintained in the base Topology Service. That is, an external link may be identified as a 10Gbps link, with a VLAN range of 1000-2000 available for dynamic aggregate stitching. However, the fact that 3 Gbps and VLAN 1051 may be already reserved at any one instant in time is generally not maintained in the Topology Service. The Topology Service is generally utilized to seed the Stitching Function path computation and workflow processes. It is through these real-time processes that the slice specific multi-aggregate resource identification and provisioning actions occur. Additional information on this is provided in the discussion of the Stitching Function step.

-Step 1-Path Computation Request: This is the step where the client software requests a Stitching Path Computation. In this case, the client desires a slice which includes slivers on AM A and AM C. The purpose of this path computation request is to identify what other aggregates and/or external networks are needed in order to stitch these two aggregates together.

-Step 2: Path Computation: In this step the Stitching Path Computation function retrieves the topology data from the Topology Service and performs a path computation. It should be noted that the result calculated here does not take into account the real-time status of bandwidth, VLANs, or other dynamically provisioned resources which may not be available due to other slice usage. The purpose of this path computation phase is to identify one or more paths that may work based on the non dynamic features of the topology like basic

connectivity, total available bandwidth, VLAN ranges, identification of situations which will require vlan translation, and other constraints.

-Step 3-Path Computation Response: In this step the Path Computation function returns a path result of AM A: AM B: EN 1: AM C. Based on the result, the client could adjust the path request and repeat step 1.

-Step 4-Create Sliver Request: In this step the client asks the aggregates of interest to instantiate the slivers via the CreateSliver command. Some clients may choose to request stitching services in before regular aggregate sliver resources. This would result in step 4 following step 10 in this diagram.

Step 5 - Stitching Request: In this step the client asks the Stitching Workflow function to coordinate the stitching in accordance with the path computation results obtained in step 3. It should be noted that some clients may desire to change the order of Steps 4 and 5. In addition, the path computation results obtained in Step 3 can be provided to the Stitching Workflow function as a "strict" or a "loose" request. That is, while the client definitely wants AM A and AM C to be stitched together in the end, it may decide to give the Stitching Workflow the permission to find paths other then AM A: AM B: EN 1: AM C if it that is needed.

Step 6-Stitching Workflow: This step represents the computation and workflow management tasks for which the Stitching Function is responsible. The Stitching Function component is where the real intelligence is located to enable multi-aggregate network stitching. For the topology view shown here, we can see there are two possible network paths to stitch together AM A and AM C. This function will need to be able to look at a topology view, identify potential stitching paths, execute a workflow which interacts with the individual Aggregate Managers to identify the compatible set of network resources to accomplish the specific network stitching request, and then go back to the selected list of AMs to instantiate the network stitching configurations. The interactions between the Stitching Function and the Aggregate Managers will be via the GENI AM API. However some network stitching extensions would be required to handle things like stitching specific resource queries, two phase commit operations, and stitching operations/status interfaces. It should also be noted that the topology view may include an external network resource in between two GENI Aggregates. In this situation, the Stitching Function will need to know how to interact with that resource dynamic provisioning API (e.g. ION). If there is a statically configured external network resource in between two GENI Aggregates, then that generally will not be reflected in the topology view. An example of this would be a regional network which may have mapped a range of 10 VLANs thru to a GENI Aggregate. In this situation, we can just reflect that VLAN range in the peering GENI Aggregates and ignore the external network in the path as far as topology views or dynamic stitching operations are concerned.

Step 7-10: These are the interactions with the individual Aggregate Managers. As mentioned in the Step 6 discussion, some extensions will be needed to the current GENI AM API to accommodate network stitching specific functions. In addition to the GENI Aggregate Managers, there may be an external network resource in the topology. In this situation, the Stitching Function (via the associated workflow) will need to have the ability to interact with that external resource in accordance with its API. The sequence shown in Figure 3a is intended to represent an example where the External Network 1 has the ability to do VLAN translation, so it is the last in the workflow to receive the stitching request.

Figure 3a depicts the interactions between the Stitching Function and the AMs in what is typically referred to as "Tree" model. That is, the Stitching Workflow Function takes responsibility for talking to all the AMs individually, and then acting on all the received information to find a solution which works for the specific stitching activity. Another option is the "Chain" model where AMs may talk directly to each other and information sharing and processing is done in a chain from AM to AM. In the Chain model, the individual AMs will need to also have some of the Stitching Function embedded. A third option is the "Hybrid" model, where the Stitching Function will initiate communications with multiple AMs, but there may also be CHAIN regions where there is direct AM to AM communications. Figures 3b-3d depict some of these other models. The intent is that a single set of Stitching Architecture Components can be utilized to support all of these modes. Below is a listing of the versions of Figure 3:

- Figure 3a Tree Model - Client Tools Initiated
- Figure 3b Tree Model - Aggregate Manager Initiated
- Figure 3c Chain Model
- Figure 3d Hybrid Model - Tree and Chain

The figures include descriptive text, so a description similar to the above is not provided for all. However, a few notes are included below regarding the main differences and features of Figures 3b-d.

Figure 3B:
The main difference between Figure 3b and 3a, is that in Figure 3b the AM initiates the Stitching Workflow processing.

Figure 3C:
This figure shows a chain mode stitching process. In this mode of operation peer aggregates will talk directly with each other. This mode will still rely on a Topology Service and Path Computation function to determine who the members of the chain will be.

Figure 3D:
This figure shows a hybrid model, where the Stitching Workflow process realizes it can stitch together multiple AMs and External Networks (ENs) via initiation of a chain process at AM B.

5.0 GENI Network Stitching - Summary

The example network stitching architecture discussed in Section 4 was intended to describe one possible mechanism to realize these functions within GENI. It is by no means the only approach. However, it is likely that a scalable and robust network stitching solution will require the following components in some form:

- Stitching Resource Element
- Common Network Stitching Topology Schema
- Stitching Topology Service
- Stitching Path Computation Function
- Stitching Workflow Function
- GENI AM API Stitching Extensions

Once these components are available, there will be enough information to decide WHAT to do to respond to a specific slice related network stitching request.

The HOW to do stitching can be accomplished in multiple ways and is generally not as hard to figure out as the WHAT to do. The HOW questions and techniques revolve around issues like:

- Should the Network Stitching function be embedded in the Clearinghouse, the individual Aggregate Managers, the client software, or some other function?
- Should we use a "Tree" or "Chain", or "Hybrid" model for multi-aggregate stitching?

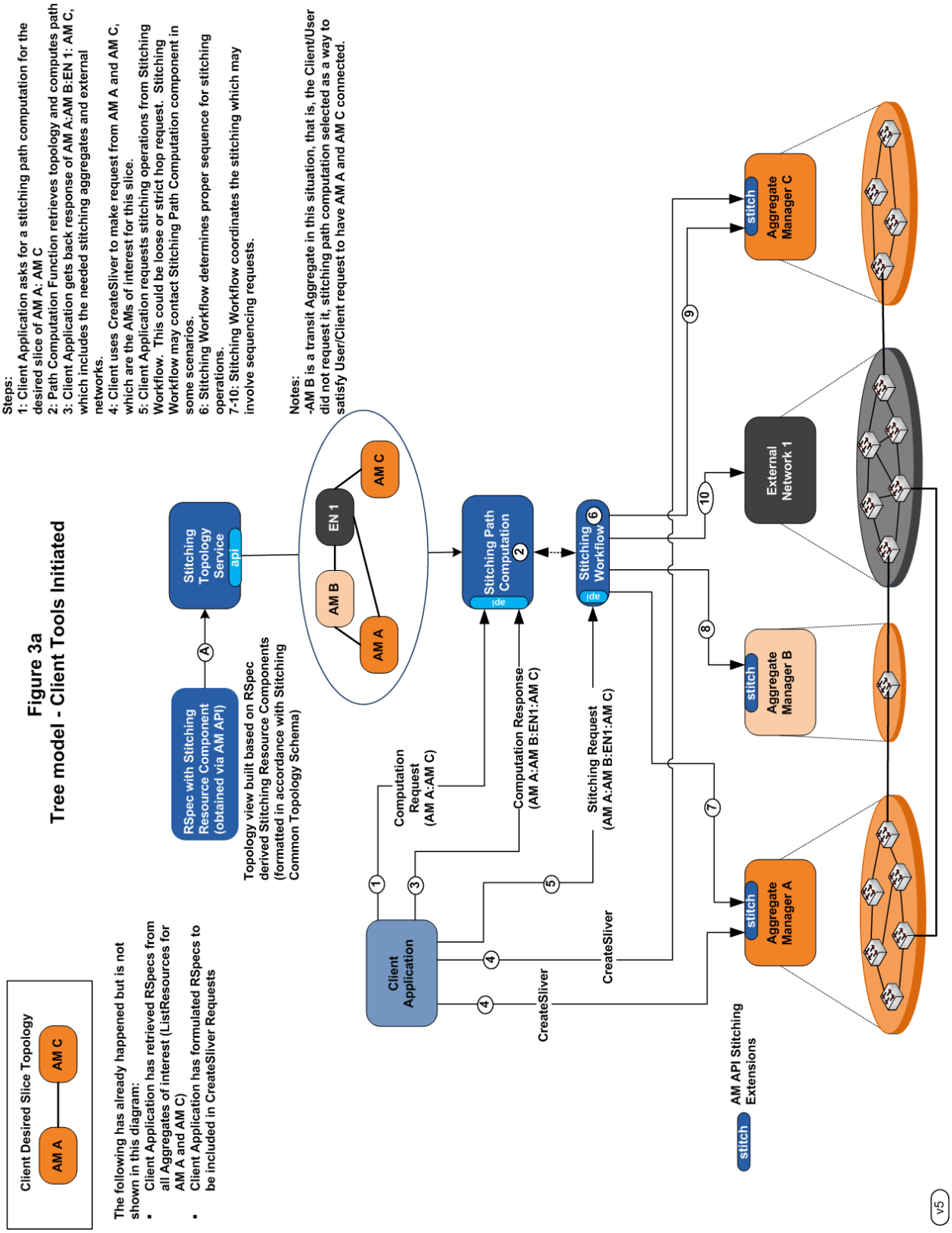


Figure 3a
Tree model - Client Tools Initiated
Client Desired Slice Topology
AM A
AM C
The following has already happened but is not shown in this diagram:
Client Application has retrieved RSpecs from all Aggregates of interest (ListResources for AM A and AM C)
Client Application has formulated RSpecs to be included in CreateSliver Requests
Steps:
1: Client Application asks for a stitching path computation for the desired slice of AM A: AM C
2: Path Computation Function retrieves topology and computes path
3: Client Application gets back response of AM A:AM B:EN 1: AM C, which includes the needed stitching aggregates and external networks.
4: Client uses CreateSliver to make request from AM A and AM C, which are the AMs of interest for this slice.
5: Client Application requests stitching operations from Stitching Workflow. This could be loose or strict hop request. Stitching Workflow may contact Stitching Path Computation component in some scenarios.
6: Stitching Workflow determines proper sequence for stitching operations.
7-10: Stitching Workflow coordinates the stitching which may involve sequencing requests.
Notes:
-AM B is a transit Aggregate in this situation, that is, the Client/User did not request it, stitching path computation selected as a way to satisfy User/Client request to have AM A and AM C connected.
RSpec with Stitching Resource Component (obtained via AM API)
Stitching Topology Service
api
Topology view built based on RSpec derived Stitching Resource Components (formatted in accordance with Stitching Common Topology Schema)
AM A
AM B
EN 1
AM C
Client Application
Computation Request (AM A:AM C)
Computation Response (AM A:AM B:EN1:AM C)
Stitching Path Computation
api
Stitching Request (AM A:AM B:EN1:AM C)
Stitching Workflow
api
CreateSliver
CreateSliver
stitch
Aggregate Manager A
stitch
Aggregate Manager B
External Network 1
stitch
Aggregate Manager C
AM API Stitching Extensions
v5

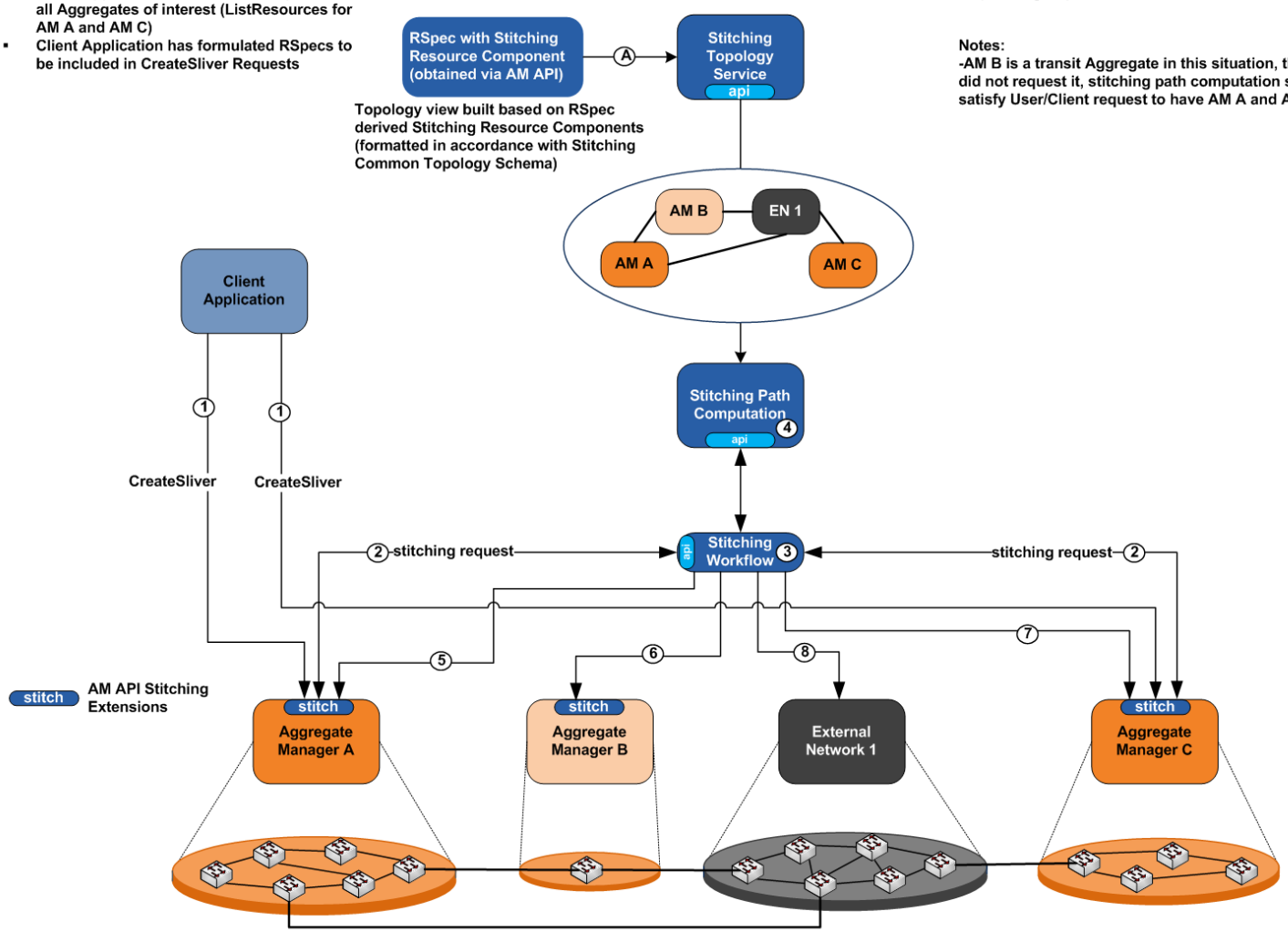
Client Desired Slice Topology

AM A — AM C

Figure 3b
Tree model - Aggregate Manager Initiated

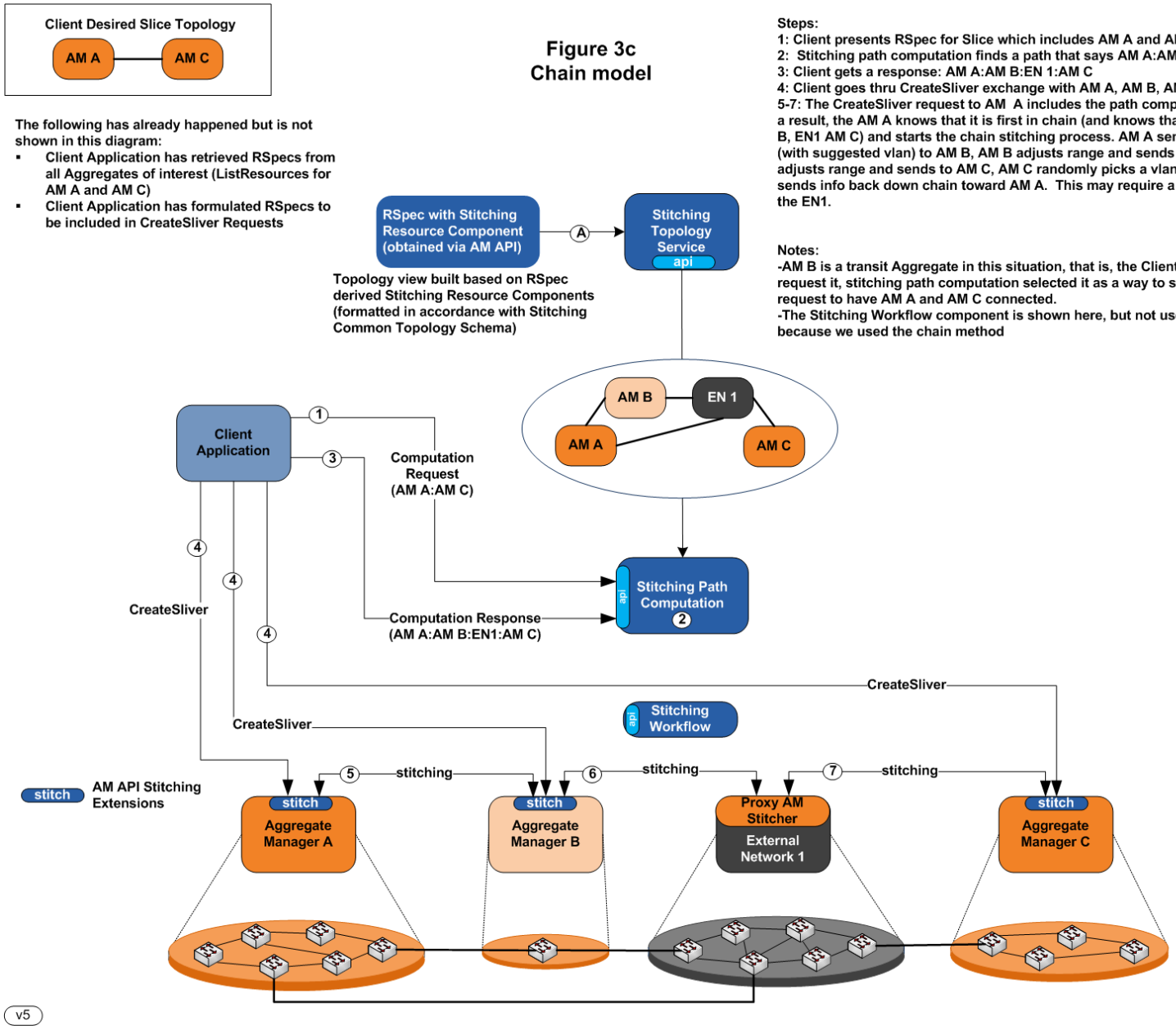
The following has already happened but is not shown in this diagram:

- Client Application has retrieved RSpecs from all Aggregates of interest (ListResources for AM A and AM C)
- Client Application has formulated RSpecs to be included in CreateSliver Requests



Steps:
1: Client uses CreateSliver to make request from AM A and AM C, which are the AMs of interest for this slice.
2: AM A and AM C realize they are not directly connected to each other, so ask for help from Stitching Workflow process
3-4: Stitching Workflow contacts Stitching Path Computation and gets a result that says use AM A:AM B:EN 1:AM C
5-8: Stitching Workflow coordinates the stitching which may involve sequencing requests.

Notes:
-AM B is a transit Aggregate in this situation, that is, the Client/User did not request it, stitching path computation selected it as a way to satisfy User/Client request to have AM A and AM C connected.


Client Desired Slice Topology
AM A
AM C
Figure 3c
Chain model
The following has already happened but is not shown in this diagram:
Client Application has retrieved RSpecs from all Aggregates of interest (ListResources for AM A and AM C)
Client Application has formulated RSpecs to be included in CreateSliver Requests
Steps:
1: Client presents RSpec for Slice which includes AM A and AM C
2: Stitching path computation finds a path that says AM A:AM B:EN 1:AM C
3: Client gets a response: AM A:AM B:EN 1:AM C
4: Client goes thru CreateSliver exchange with AM A, AM B, AM C
5-7: The CreateSliver request to AM A includes the path computation results. As a result, the AM A knows that it is first in chain (and knows that path is AM A, AM B, EN1 AM C) and starts the chain stitching process. AM A sends a range of vlans (with suggested vlan) to AM B, AM B adjusts range and sends to EN 1, EN1 adjusts range and sends to AM C, AM C randomly picks a vlan from range and sends info back down chain toward AM A. This may require a Proxy AM to cover the EN1.
Notes:
-AM B is a transit Aggregate in this situation, that is, the Client/User did not request it, stitching path computation selected it as a way to satisfy User/Client request to have AM A and AM C connected.
-The Stitching Workflow component is shown here, but not used in this scenario, because we used the chain method
RSpec with Stitching Resource Component (obtained via AM API)
A
Stitching Topology Service
api
Topology view built based on RSpec derived Stitching Resource Components (formatted in accordance with Stitching Common Topology Schema)
AM B
EN 1
AM A
AM C
Client Application
1
3
Computation Request (AM A:AM C)
4
4
4
CreateSliver
Computation Response (AM A:AM B:EN1:AM C)
Stitching Path Computation
2
CreateSliver
CreateSliver
Stitching Workflow
stitch
AM API Stitching Extensions
5
stitching
6
stitching
7
stitching
stitch
Aggregate Manager A
stitch
Aggregate Manager B
Proxy AM Stitcher
External Network 1
stitch
Aggregate Manager C
v5

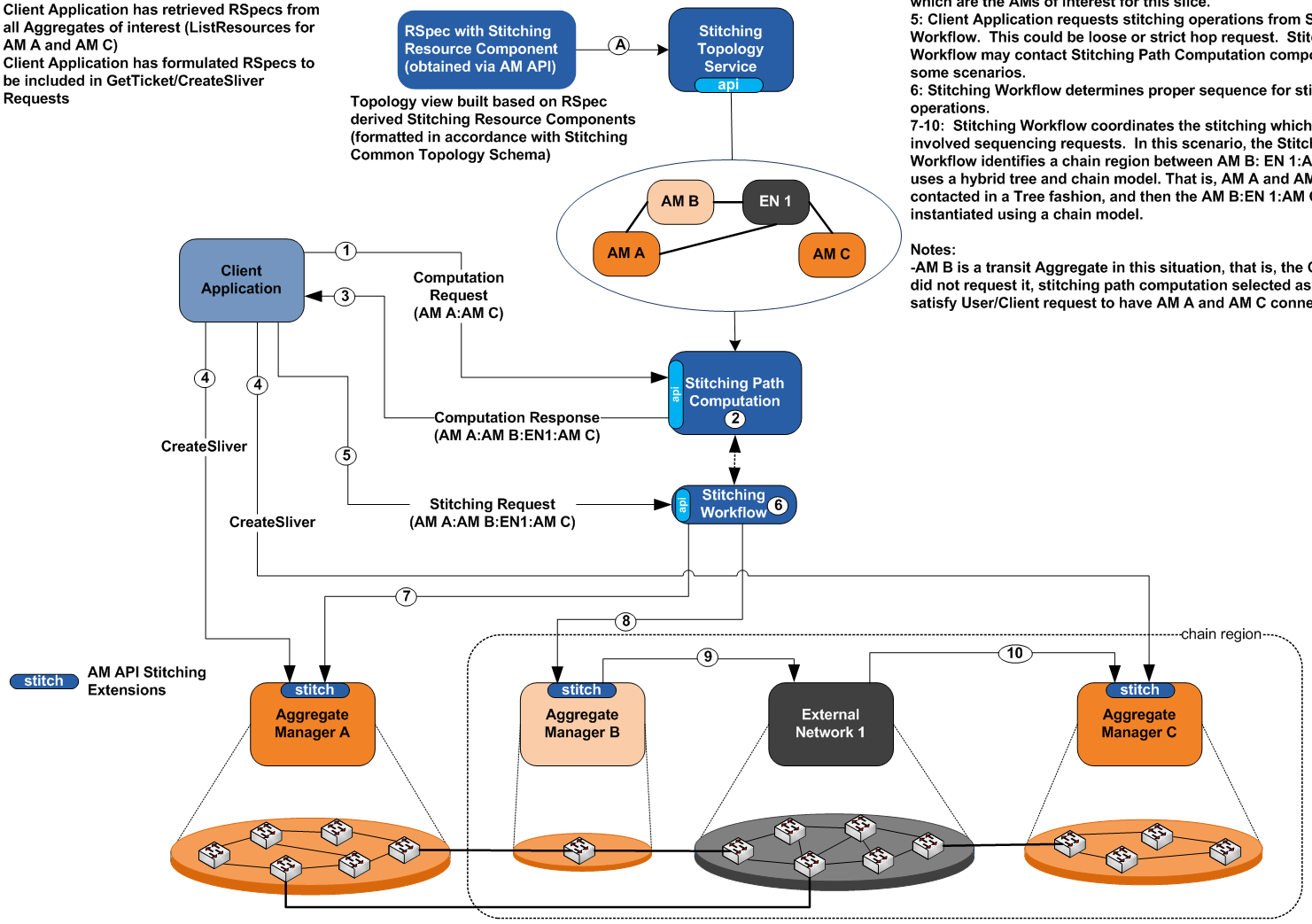
Client Desired Slice Topology

AM A — AM C

The following has already happened but not shown in this diagram:

- Client Application has retrieved RSpecs from all Aggregates of interest (ListResources for AM A and AM C)
- Client Application has formulated RSpecs to be included in GetTicket/CreateSliver Requests

Figure 3d
Hybrid Stitching Example
Tree and Chain



Steps:
1: Client Application asks for a stitching path computation for the desired slice of AM A: AM C
2: Path Computation Function retrieves topology and computes path
3: Client Application gets back response of AM A:AM B:EN 1: AM C, which includes the needed stitching aggregates and external networks.
4: Client use CreateSliver to make request from AM A and AM C, which are the AMs of interest for this slice.
5: Client Application requests stitching operations from Stitching Workflow. This could be loose or strict hop request. Stitching Workflow may contact Stitching Path Computation component in some scenarios.
6: Stitching Workflow determines proper sequence for stitching operations.
7-10: Stitching Workflow coordinates the stitching which may involved sequencing requests. In this scenario, the Stitching Workflow identifies a chain region between AM B: EN 1:AM C and uses a hybrid tree and chain model. That is, AM A and AM B are contacted in a Tree fashion, and then the AM B:EN 1:AM C path is instantiated using a chain model.

Notes:
-AM B is a transit Aggregate in this situation, that is, the Client/User did not request it, stitching path computation selected as a way to satisfy User/Client request to have AM A and AM C connected.

v5