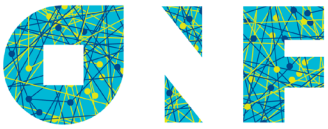

OPEN NETWORKING
FOUNDATION

Orchestration:
A More Holistic View

January 2017

ONF TR-540

Disclaimer

Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
www.opennetworking.org

Orchestration: A more holistic view

The aim of this paper is to expand a commonly used concept of orchestration and wants the reader to acknowledge (or at least be mindful of) its wider features. It offers the community a brief discussion of differences in the way the term is used, and why and how these differences matter. This paper explores the overall functionality that must be provided, whether encompassed in a single large-scale orchestration wrapper or partitioned into several sub-functions, of which only one component is called an orchestrator.

Orchestration is a central aspect of the current industry conversation about network evolution, but the meaning and scope of orchestration is often only implied, and various communities do not always understand the implications in the same way. Its term is used everywhere throughout the community, but often is taken as implicitly understood.

All understandings of orchestration include the idea of automatically selecting resources to satisfy client demands. The term *client* represents any entity that can use resources or services from a provider. Client and provider may or may not exist in separate business or administrative domains. An orchestration activity may also use resources and services for its own purposes.

However, common usage differs, explicitly or implicitly, in several areas discussed below.

Orchestration is sometimes understood to refer only to services and often only to end-to-end services. Orchestration is sometimes understood to exist at only a single Management/Control level.

However, a service is in fact the right to use, or the actual use of, some set of resources. There is no fundamental distinction between service provisioning and resource provisioning. Only the level of abstraction and detail vary, depending on the scope and perspective of the orchestration entity.

With respect to the end-to-end qualifier, from the viewpoint of the orchestrator involved, all services are end-to-end. When a higher-level orchestrator disaggregates a service that spans its endpoints, it hands off segments to subordinate orchestrators, each of whose endpoints are stitched together by the superordinate orchestrator. Beyond the endpoints within its own scope, an orchestrator has little or no visibility.

To illustrate, the service provided by a carriers' carrier is clearly only a fragment of a wider service. The carriers' carrier orchestrator rightly perceives this as an end-to-end service, while a superordinate orchestrator perceives the fragment as a possibly configurable resource or set of resources.

In some cases, an orchestrator's endpoint may be a business domain boundary, requiring the pre-establishment of a contractual agreement for SLA and billing. The exchange of information regarding usage, performance and troubleshooting is normally required even for endpoints within a single administrative domain.

In some formulations, a function that doesn't span some form of domain boundary is not regarded as orchestration.

Services normally require decomposition into components, each of which impinges on others at boundaries that must be known and coordinated. An orchestrator must necessarily be able to coordinate services and resources across its scope (including at its own edge-points), whether within or across business boundaries.

Given that orchestration is recursive, insistence that the lowest level of recursion was not orchestration would imply that there necessarily exists some level beyond which we cannot go, and that it matters. But the bottom of an abstraction stack is a matter of choice. Even if the scope of an orchestrator is a single network element (NE), for example, there will be a further abstract hierarchy within the NE that decomposes box-level services into, for example, cards and chips.

Orchestration is sometimes understood to be driven by BSS-OSS inputs.

The BSS-OSS assumption presupposes the continued existence of BSS-OSS much as they

are today, and hence an upper limit to recursive orchestration. If BSS-OSS responsibilities are instead viewed as collections of functional components, they are themselves compatible with the idea of orchestration: the use of various (abstract) resources and functions to achieve desired outcomes. Orchestration is part of a general pattern that is repeated recursively throughout the solution.

Orchestration is sometimes understood to be independent per service instance.

Improving the efficiency of resource use is a continuing objective of current network management approaches. In a loaded network, the probability of service provisioning or operational failure can be kept down by allowing for the dynamic rearrangement of contending services, meaning that orchestration needs to consider all services together. Re-balancing services is a vital, although admittedly complex, task.

Feedback, dynamic response (including response to signaling events), optimization and policy are often considered to be separate from orchestration.

All concepts of orchestration assume that initial service provisioning takes resource availability (state) into account. Discussions of orchestration usually include mention of service assurance, recognizing the need to maintain services in light of dynamically changing resource state. What is often omitted is the need for resource loading optimization on a continuing basis, and the necessary incorporation of policy that defines what is meant by optimization in the presence of tradeoffs.

Many treatments of orchestration classify network state feedback into a separate category, Analytics, and guidance into a separate category, Policy. There is no question that resource feedback is vital, and that choices should be determined by policy. In fact, the choice of feedback and its analysis should itself be driven by policy, as well as the choices made to allocate services across available resources.

If orchestration is deemed to be little more than route discovery or path computation on an over-engineered network, feedback and policy will have to be provided by other tightly-integrated components. Further, the interfaces between these components will need to be wide and rich, as Analytics evaluates information in light of network and service commitments known to Orchestration, makes decisions according to Policy, and proposes re-balancing actions.

As to re-balancing: if optimization is thought of as adjusting knobs on the network, it will be necessary to decide which knobs to adjust, and by how much. This will require prediction of the consequences (simulation) before decisions are finalized, a thought experiment of trial resource selection, prediction of the resulting state, and policy to evaluate the result and guide the next trial iteration. Even the narrowest view of orchestration implies policy intelligence in selecting the appropriate knobs and proposing how far to turn them.

> [ONF-TR521] defines orchestration: The ongoing selection and use of resources by a provider to satisfy client demands according to optimization criteria.
>
> This definition is intended to compactly encompass all of the necessary aspects of a solution, while not compelling any particular subdivision of functionality, e.g., into policy or analytics that are somehow separate.

Product view

In the overall community, orchestration is intended to capture the idea of automated service delivery. This goal requires a number of functions, however, some of which are included in the various concepts of orchestration, some of which are stated or implied to exist separately. Whatever the terminology, the important point is that all functions be present and integrated into a coherent whole.

While it is acceptable for a product to not include all necessary features for a total solution and for that product to be branded as an orchestrator, it is vital that the architecture be separated from such product branding decisions. A clear primary objective of an architecture is to define the set of necessary functions for an overall solution. Fundamentally we need to understand the functions required and their interconnection. How we then choose to label groups of functions is secondary, as long as the right interconnection of functions is in place.

Such key functions are identified on the next page.

Considerations of the Capabilities of Orchestration

Regardless of how we partition the functions among branded elements, the overall solution requires functions:

- To best offer appropriate purposeful capability that:
  - Employs policy to guide all decisions.
  - Offers flexible arrangement of tasks to enable operational variety and evolution
- To offer services to clients (in client friendly terminology)
- To determine an assembly of resources to form a system that supports the service, where that system can itself can be recursively viewed as a resource for further use, and to realize that assembly/view
- To take a service intent agreement and to determine an appropriate resource/service assembly to best satisfy the agreement simultaneously with all other agreements where the resources/services may be provided by other subordinate/peer orchestrators, including those in third party solutions.
- To collect meaningful state information from the orchestrated resource/service assembly, analyze it, and use the results to guide further decisions.
- To aim to best satisfy all service demands simultaneously, adjusting, reactively and proactively, the realizations to deal with changes in the environment on an ongoing basis and to escalate where unable to satisfy demand as appropriate based upon policy
- To optimize resource utilization to maximize return on resource investment
- To feed billing and regulatory systems with necessary information to enable them to perform their functions

It would be nice if the world converged on a single common definition of orchestration, but what is more important is that, whenever a partial definition is used, there is also a checklist that encourages consideration, somehow, of the other essential factors.

This document comes out of the ONF Architecture group and urges the community within and beyond ONF to realize and recognize the broader features of orchestration during all discussions and especially when publishing papers. The ONF architecture members welcome comments and thoughtful additions to this list of considerations.

References

[ONF-TR521] TR-521, SDN architecture 1.1, 2016, https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-521_SDN_Architecture_issue_1.1.pdf