

Orchestration vs. Choreography

Asked 14 years, 1 month ago Modified 1 year, 9 months ago

Viewed 155k times



282

What are the differences between service orchestration and service choreography from an intra-organization point of view.



soa

composition

service-composition



Share

Improve this question

Follow

edited Dec 7, 2017 at 3:44



Andrei

7,607 ● 8 ● 36 ● 64

asked Nov 8, 2010 at 19:26



PetrosB

4,154 ● 5 ● 23 ● 21

11 Answers

Sorted by:

Highest score (default)



550

Basic technologies (such as XML, SOAP, WSDL) provide means to describe, locate, and invoke services as an entity in its own right. However, these technologies do not give a rich behavioral detail about the role of the service in more complex collaboration. This collaboration includes a sequence of activities and relationships between



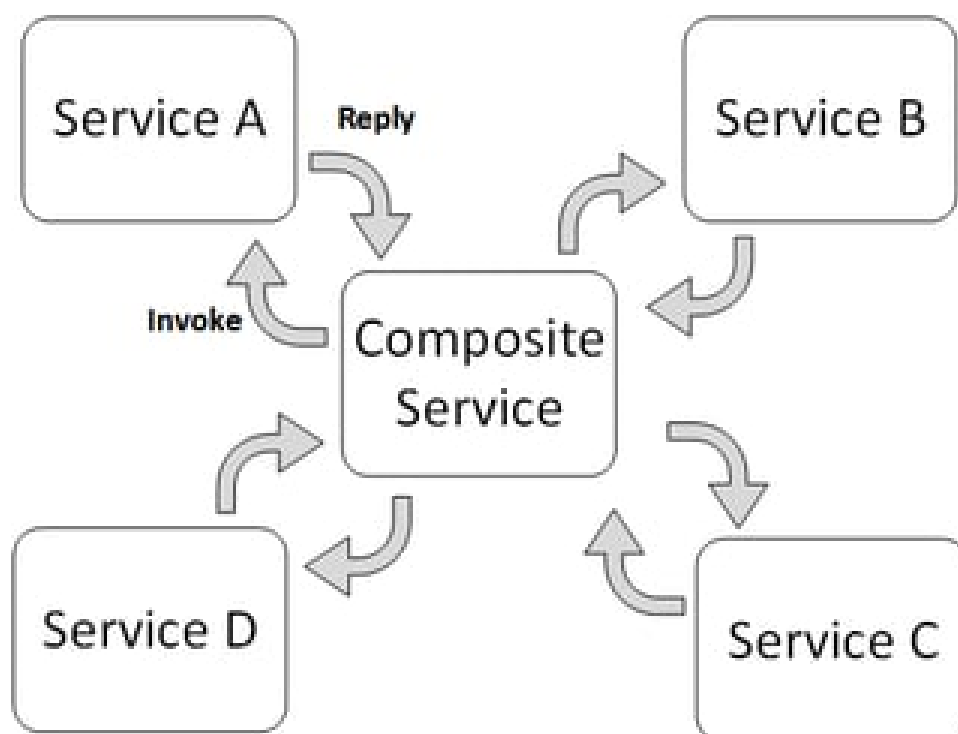


activities, which build the business process. There are two ways to build this process: service orchestration and service choreography.

Service orchestration

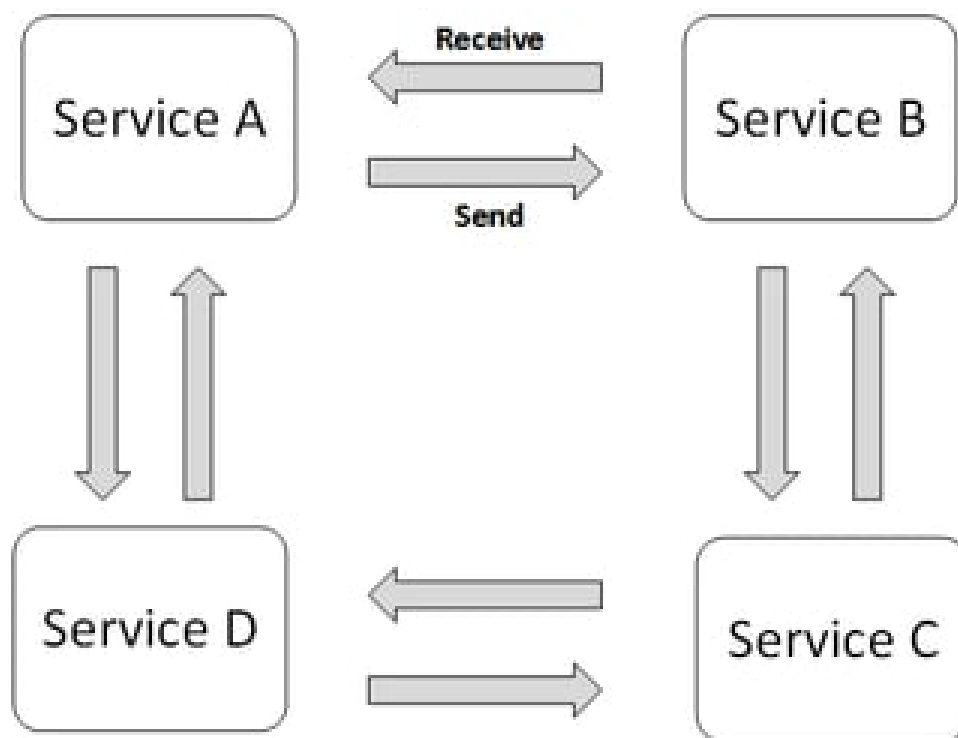
Service orchestration represents a single centralized executable business process (the orchestrator) that coordinates the interaction among different services. The orchestrator is responsible for invoking and combining the services.

The relationship between all the participating services are described by a single endpoint (i.e., the composite service). The orchestration includes the management of transactions between individual services. Orchestration employs a centralized approach for service composition.



Service Choreography

Service choreography is a global description of the participating services, which is defined by exchange of messages, rules of interaction and agreements between two or more endpoints. Choreography employs a decentralized approach for service composition.



The choreography describes the interactions between multiple services, where as orchestration represents control from one party's perspective. This means that a **choreography** differs from an **orchestration** with respect to where the logic that controls the interactions between the services involved should reside.

Share Improve this answer

Follow

edited Mar 8, 2021 at 9:17



Benny Bottema

11.5k ● 10 ● 75 ● 102

answered Apr 22, 2015 at 21:01



Andrei

7,607 ● 8 ● 36 ● 64

15 Fantastic images to show the two! Where did you get them?
– [David Mann](#) Apr 14, 2017 at 15:55 ✎

12 @DavidMann Many thanks for your comment, and please upvote if you think it was useful to you. I made these diagrams in Visio, and as inspiration I used some literature in service composition. However, I realised that this answer was 2 years ago when I just started reading about service composition. I will update this answer with references and expand on the proprieties of both – [Andrei](#) May 26, 2017 at 17:19

1 @Andrei: Couldn't have been more simpler than this.
– [Anshul Nigam](#) May 24, 2018 at 6:04

Is it reasonable to mix orchestration and choreography? e.g. have orchestration for the core synchronous workflow, but then some choreography to stream async events back to the source capabilities (microservices). In my scenario, this approach might save me having to do saga/state-machine and compensation logic. – [Ryan.Bartsch](#) Oct 14, 2019 at 22:21

8 Some readers may infer from your orchestration diagram that orchestration means synchronous invocations from the controlling service to providing services. I'd like to clarify that Invoke-Reply communication can also be accomplished using asynchronous methods, incl. via a message broker.
– [Christoph](#) Apr 13, 2020 at 11:07



Service orchestration: you put together several services by a fixed logic. This logic is described at a single place.



You can imagine a team of people with a manager doing micro-management. The manager precisely tells what, when and who should do. The team members do not care of the entire goal of the job, the manager combines the outputs into a single deliverable. A practical example is a BPEL process. BPEL process contains the logic, can invoke several services and combine their responses into a single service response.

Service choreography: the decision logic is distributed, with no centralized point. You can imagine a home, where everybody aims for the common good and works pro-actively without micro-management. Or you can imagine a human body, where different members are interdependent and work for the common goal. A practical example is event driven processing, where an agent is activated by an event and does its job. All the agents make a system together. There is no centralized logic. Choreography possibilities may go farther beyond orchestration as it is more aligned with the real world.

My opinion is that we do not need to distinguish much between these two, as we need to focus on the business logic. Where a single point of logic does the job, we do orchestration. Where a problem cannot be covered by a centralized logic, we are forced to choreography anyway. That is why we often come accross orchestration in IT, whereas choreograhly remains more an academical concept and a subject for research. And very often we do choreography without actualy knowing it, as in the real world.

answered Sep 7, 2016 at 8:11

**KarelHusa**

2,253 ● 20 ● 30

**30**

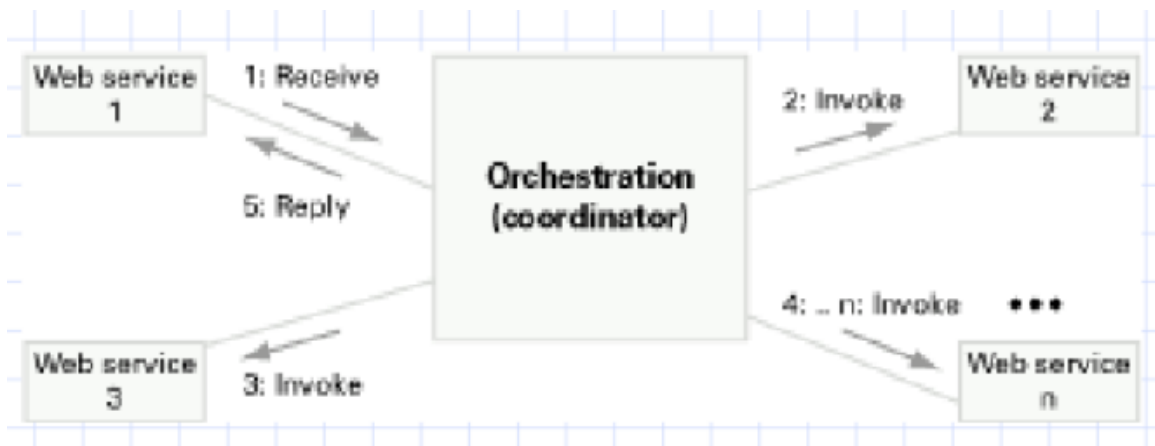
Since the thread is old but still writing to it for those who will stumbled here in search of this question as I did. This is much debated question in [Service-oriented architecture](#) (SOA) which needs much cleaner explanation for beginners.



Orchestration: Executable Process

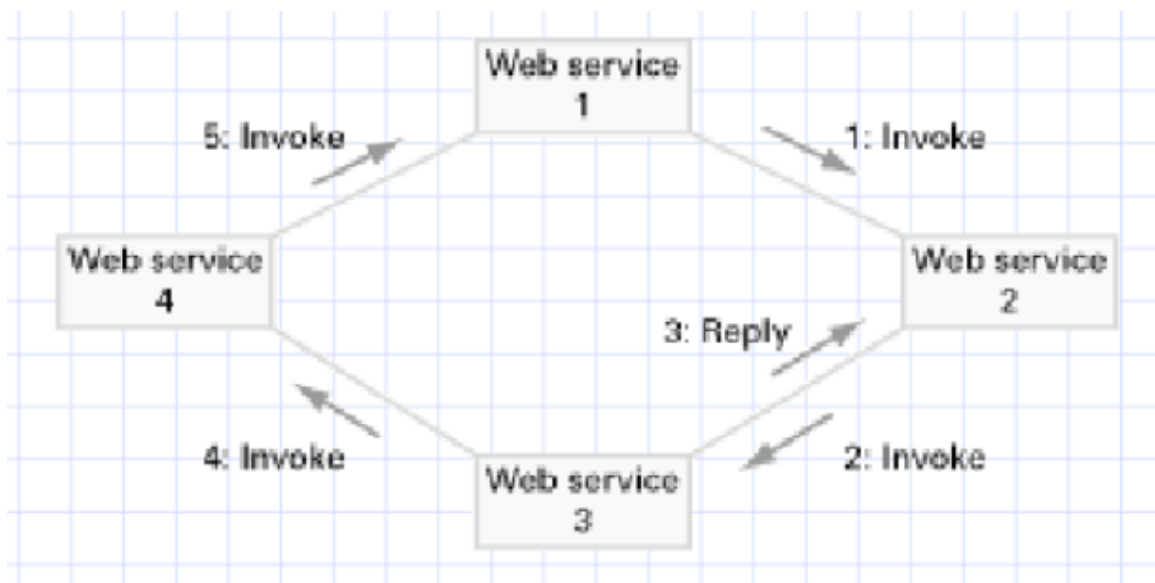


- Used in private business processes
- A central process (which can be another Web service) takes control of the involved Web services and coordinates the execution of different operations on the Web services involved in the operation
- The involved Web services do not "know" (and do not need to know) that they are involved in a composition process and that they are taking part in a higher-level business process.
- Only the central coordinator of the orchestration is aware of this goal, so the orchestration is centralized with explicit definitions of operations and the order of invocation of Web services.



Choreography: Multi-party Collaboration

- Choreography, in contrast, does not rely on a central coordinator. Rather, each Web service involved in the choreography knows exactly when to execute its operations and with whom to interact. Choreography is a collaborative effort focusing on the exchange of messages in public business processes.
- All participants in the choreography need to be aware of the business process, operations to execute, messages to exchange, and the timing of message exchanges.



Choreography vs. Orchestration

- From the perspective of composing Web services to execute business processes, orchestration is a more flexible paradigm and has the following advantages over choreography:
- The coordination of component processes is centrally managed by a known coordinator.
- Web services can be incorporated without their being aware that they are taking part in a larger business process.
- Alternative scenarios can be put in place in case faults occur.

Share Improve this answer

Follow

edited Jan 12, 2016 at 3:10



Marco Roy

5,181 ● 7 ● 42 ● 64

answered Oct 24, 2015 at 10:13



Cyclotron3x3

2,219 ● 1 ● 27 ● 41

2 Actually, choreography usually does rely on a central coordinator - and usually that coordinator is some form of distributed message broker. Not using something like a message broker would tie your services together in very inflexible ways leading to brittleness and low reusability.

– [Rodney P. Barbati](#) Oct 3, 2019 at 18:44

3 Don't follow @RodneyP.Barbati. How is an asynchronous message bus to alleviate dependency brittleness equivalent to a central coordinator. The message broker is completely dumb and unaware of any processes that might rely on it.

– [Benny Bottema](#) Mar 8, 2021 at 9:20

@Benny Bottema - because each service/function involved in the process is sending messages out that trigger the next service/function in the process chain. They are not calling the next service/function directly. – [Rodney P. Barbati](#) Sep 10, 2021 at 15:45

- 3 Except the bus doesn't coordinate anything, quite the opposite: the events and consuming services form a choreography of which the bus has no notion whatsoever. solace.com/blog/microservices-choreography-vs-orchestration – [Benny Bottema](#) Sep 10, 2021 at 15:57
-



26



Services can be distinguished between atomic services and services composed of other services. Such compositions are called "orchestration". Sometimes workflow, sometimes business process. For instance, BPEL is an orchestration language, but calls itself "business process execution language".

There is no requirement that services need to be hierarchically composed. That means, two services may talk to each other. The protocol running between them is called "choreography". It may be two services, but usually, there are more than two services involved. Each service in a choreography may be seen as orchestrator of the partner services. Each service taking part in a choreography may be realized as orchestration/workflow/process.

An orchestration shows the complete behavior of each service whereas the choreography combines the interface behavior descriptions of each service.

A good scientific article distinguishing choreography, interface behavior, provider behavior, and orchestration is the following one: [Dijkman, R. & Dumas, M. Service-oriented Design: A Multi-viewpoint Approach International Journal of Cooperative Information Systems, 2004, 13, 337-368](#)

Share Improve this answer

edited Feb 22, 2012 at 21:31

Follow

answered Oct 6, 2011 at 14:37



kopper

20.4k ● 16 ● 128 ● 173



18



Andrei and others did a good job explaining what is orchestration and what is choreography. For the software architect choosing between these two alternatives, it is also important to compare them with respect to different qualities.



Orchestration pluses over choreography



- **Reliability:** Orchestration platforms have built-in support for error handling and transaction management (compensating transactions). In choreography, custom-developed workflow and error handling tends to be more error-prone. Also, choreography is often event-driven and much of the processing is asynchronous. Therefore,

choreography may require undo/correction events that add complexity to the solution.

- **Modifiability:** Creating and changing process workflows and complex service compositions is easier in the visual BPM tools found in orchestration platforms. You gain in "process visibility".

Choreography pluses over orchestration

- **Performance:** Orchestration incurs a performance overhead due to workflow script interpretation and the additional layer of the orchestration platform itself.
- **Cost:** Choreography does not require additional middleware or language, which have associated learning curves and governance burden.

EDIT

An orchestration solution might introduce a SPOF if the orchestrator element doesn't employ a mechanism for high availability. Thanks @Deepak for pointing this out in a comment.

Share Improve this answer

edited Jul 7, 2021 at 15:36


Follow

answered Feb 11, 2016 at 11:45



Paulo Merson

14.4k ● 10 ● 88 ● 76

-
- 5 Except choreography does require additional middleware. Requirements (tasks) need to be matched to different nodes. Afterwards the choreography is deployed, enacted, monitored and adapted. All this management requires tools that are usually provided by a middleware. – [Andrei](#) Feb 14, 2016 at 23:01 
-
- 5 Doesn't orchestration bring a disadvantage of single point of failure which is not the case with Choreography? – [Deepak](#) Apr 14, 2020 at 15:15
-
- 1 @Deepak, A process engine could be using a replicated database for it's state. Fallover would not be too hard to do in such a setup. In addition, there can be multiple engines (just only one per process instance, which you would also have for choreography - except that choreography doesn't store per design (individual components could/can)). – [Paul de Vrieze](#) Nov 19, 2020 at 16:59
-
- 4 By far the biggest plus of an orchestrator is the transparency of the process whereas with choreography the process flow is often not clear at all unless there is comprehensive documentation available (which often there is not). For example event driven processes (choreography) have great advantages, but damn if it isn't a complete puzzle sometimes to figure out how the process flows. – [Benny Bottema](#) Mar 8, 2021 at 9:22
-
- 1 not to mention when the service composition in choreography gets huge enough you'll need reconciliation mechanisms to ensure system correctness and anomaly detection – [varun](#) Jul 7, 2021 at 8:09
-



Orchestration is useful when you have control over all the actors in a process - when they're all in one domain of control and you can dictate the flow of activities. This is of

8



course most often when you're specifying a business process that will be enacted inside one organisation that you have control over.



Choreography is a way of specifying how two or more parties - none of which has any control over the other parties' processes, or perhaps any visibility of those processes - can coordinate their activities and processes to share information and value. Use choreography when coordination across domains of control/visibility is required. You can think of choreography, in a simple scenario, as like a network protocol. It dictates acceptable patterns of requests and responses between parties.

Share Improve this answer

Follow

answered Jan 31, 2015 at 17:28



[Bergrebell](#)

4,708 ● 4 ● 44 ● 58



Another way to look at Service Orchestration vs. Choreography:

8



- **Service Orchestration: Around a Business Domain.**
- **Service Choreography: Among multiple Business Domains.**



Share Improve this answer

Follow

answered Jun 13, 2016 at 21:47



[Michael Qin](#)

665 ● 6 ● 12



6



I'd say choreography is well suited internally for highly decentralized organizations. You won't need a central business process executor. This facilitates independent growth and development by each of organization sub-units.

(I subscribe to this interpretation of orchestration vs. choreography question:

<http://geekexplains.blogspot.com/2008/07/ways-of-combining-web-services.html>)

Share Improve this answer

answered Dec 8, 2010 at 18:34

Follow



Danijel Arsenovski

519 ● 10 ● 15



2



One can use **both** choreography and orchestration is the same system as we have done in our product. Various actors performing various tasks dispatched to them are choreographed by the event pub/sub system via events they generate. For example, when a prime mover carrying a container arrives at a warehouse, that event, which is subscribed by the warehouse manager app, prompts the warehouse manager to activate the resources to unload the cargo. But when exceptions happen, such as the forklift to carry the load breaks down, that event starts an orchestration engine (a workflow processor) to orchestrate the task for various actors as per the exception handling workflow to handle the exception.

Share Improve this answer

Follow

edited Mar 16, 2022 at 16:42



cellepo


4,459 ● 4 ● 40 ● 65

answered Nov 12, 2021 at 19:43



Rafiul Ahad

21 ● 2

Yes, and this seems to be referred to in industry as the **"Hybrid"** approach of both: Searching the web with "orchestration choreography hybrid" yields numerous results of use case examples. – [cellepo](#) Mar 16, 2022 at 16:44 



1



In orchestration, there is a conductor and there are instrument players. Players play according to how conductor conducts. If conductor is replaced, the harmonic expression will be different i.e. it is still the same play (service) but with a different outcome. For example, to provide a financial arrangement proposal, the orchestration service will conduct by asking (invoking) each player (entity or utility service, e.g. credit check) to play (return results or adjust/update its playing) according to conductor's template (business rules). In choreography, there is a choreographer and there are groups of dancers. Choreography is a direction, but each group of dancers is autonomous in how to realize that direction.

Share Improve this answer

answered Sep 23, 2015 at 18:44

Follow



Bran

97 ● 1 ● 4



0



Both orchestration and choreography are two philosophies of process formalism at large scale, i.e. collaboration spaces. A collaboration space is a formal representation for an industry, for example health industry, food industry, automotive industry or So you should have some service providers at small scale able to speak some language of communication before you decide to orchestrate them or use choreography to supply some demand meaningful to end-consumer.

Orchestration typically follows what has been famous as middleware in distributed systems. Some good tooling examples within IT industry can be [Juju](#), [Zapier IFTTT](#). You have centralized control, auditory, modification and trouble shooting over services you consume.

Choreography is more decentralized than orchestration in the sense that it accepts more autonomous collaborators. For example bitcoin network is a collection of service providers called nodes which collectively provide a Defi service namely distributed ledger functionality to its service end-consumers. But this does not have any centralized authority in charge, and there is no total-view over whole network. There is just a [consensus mechanism](#) that if respected candidate service provider joins the network and participate in end-consumer service provision. Anywhere you need to attack [trusted third party](#) and its side effects such as corruption, an orchestration would not suffice.

Again any collaboration among actors in any industry can be subjected to orchestration or choreography, so I hope limiting examples only to IT industry is not misleading.

Also I can add, naming suggests analogy between the two and stage performance. In an orchestra, every instrument player is harmonized with other team members if he/she follows the single conductor. On the other hand analogous to stage dancing in choreography there is no conductor, each stage dancer only harmonize himself/herself with his/her partner (better if spoke as

peer), with no single point of harmonization leadership.
But still both present harmonization to viewers.

Share Improve this answer

edited Nov 8, 2020 at 15:06

Follow

answered Nov 8, 2020 at 14:58



shvahabi

1,308 ● 10 ● 9
