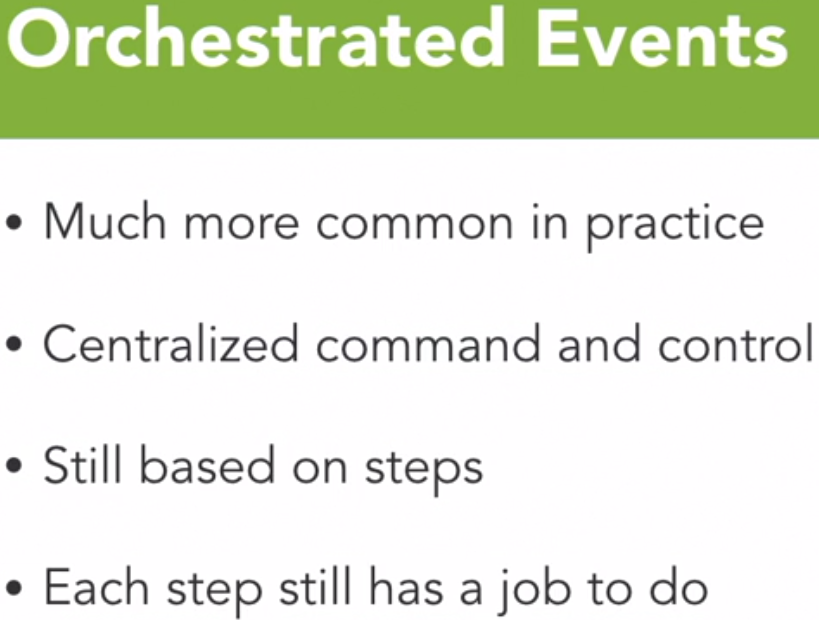
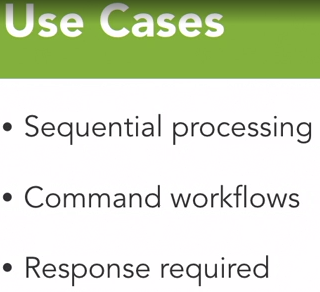
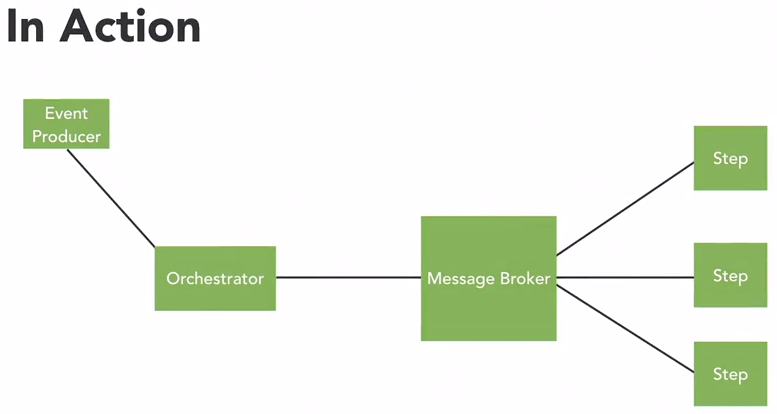


Choreographed on who need to do what. Now no one monitors as a central tower as each process is expected to take care of things (execution/error handling etc.) by themselves. Step by step process. Producer begins, a consumer receives/processes – leave another message, another consumer receives/processes – leaves another message etc.  
Every consumer does its own step. Does not coordinate with the main event producer that initiated the flow. Also not central orchestrator observing the whole flow.

Pros: High Performance, Loose Coupling  
Cons: Reduced reliability – Error handling should be taken care appropriately everywhere. Reduced observability or traceability.

Orchestrator acts as a central tower to monitor the steps executed and to start the next set of actions.

Event producer initiates the event but sends it to orchestrator.

When step1 consumer consumes, it is expected to respond back once done byt sending a message to message broker about completion. Orchestrator keeps and eye on Step1 consumer completion and then leaves a message for step 2 consumer. Step 2 consumer processes the message and then responds to Message broken which then again read by Orchestrator. While all this is happening, Orchestator would leave updates for the continuously polling Event producer.

Orchetrator will be aware of each step being performed while Event Producer keeps polling for the progress update.

Cons: Decreased performance because orchestrator is coordinating and sequencing the events only one after the other. Orchestrator needs to be like a state machine to know how to respond on each step like failure or something. Increased cost.

Pros: Increased reliability as the error handling can e controlled from one place.  
Increased observability as orcestrator has a clue on what all steps have been performed and the time taken. More logging and metrics captured can be done at orchestrator.

Hybrid of both Choreographed and Orchestrated events can be considered to balance the business needs but ultimately the central tower or orchestrator someone is to be informed about the completion. This Hybrid style would allow reduce some orchestrator overhead and gives scope for better performance.

CQRS Command Query Responsibility Segregation  
