process engine. This diagram is useful, if you want to scope the process, get a basic understanding of the flow, and talk about the main steps, but not if you want to dig into the details for discussing process improvements or even software driven support of the process.

6.2 Detailed Collaboration and Choreography

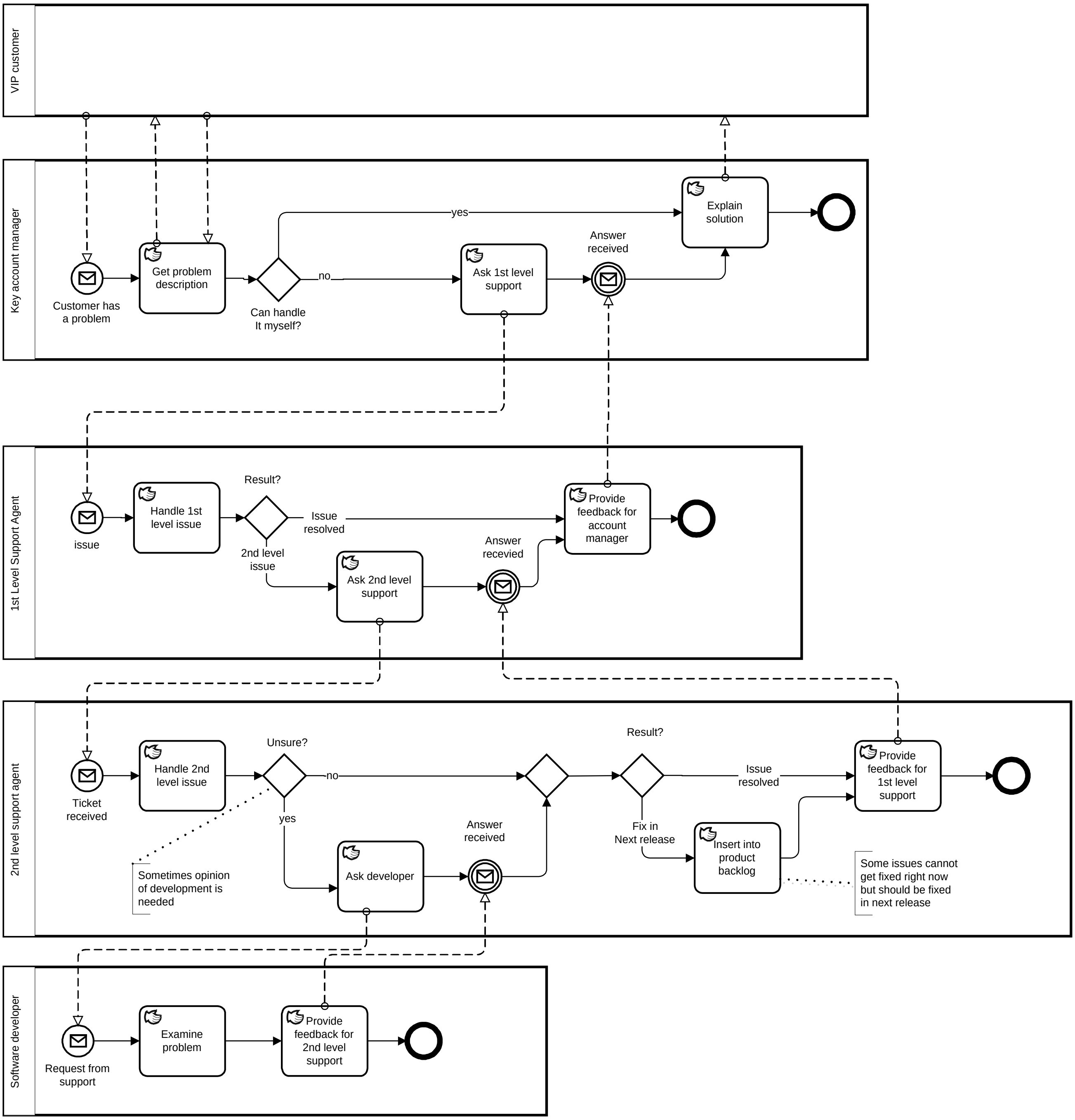

Figure 6.2: Incident Management as detailed collaboration

We can take a closer look at the ping-pong-game of account manager, support agents and software developer by switching from a single-pool-model to a collaboration diagram, as shown above. We can now see some more details about the particular processes each participant fulfills, e.g., the dialogue between the account manager and the customer for clarifying the customer's problem, or the fact that the 2nd level support agent will insert a request for a feature in the product backlog, if the current release of the software product cannot cover the customer's demand satisfactorily. We have also specified each task as manual, which means that we still think of the processes as completely human-driven with no process engine involved. This could hypothetically be the As-Is-state of the incident management before the introduction of a process engine. The next step could be to define whether we want to drive the complete collaboration by a process engine, or only parts of it. But before we discuss that matter, we can have a look at an other way of modeling such a ping-pong-game, the choreography diagram shown below. This diagram only shows the tasks that are dedicated to the communication between the different process participants, hiding all internal steps, e.g., the task that inserts a new entry into the product backlog. Note that the diagrams shown in Figure 6.1 and 6.2 have no formal connection between each other, whereas the Figure 6.2 and 6.3 have the exact same semantic model behind them and just provide different views on it. See also Annex A for an XML serialization of the underlying semantic model.

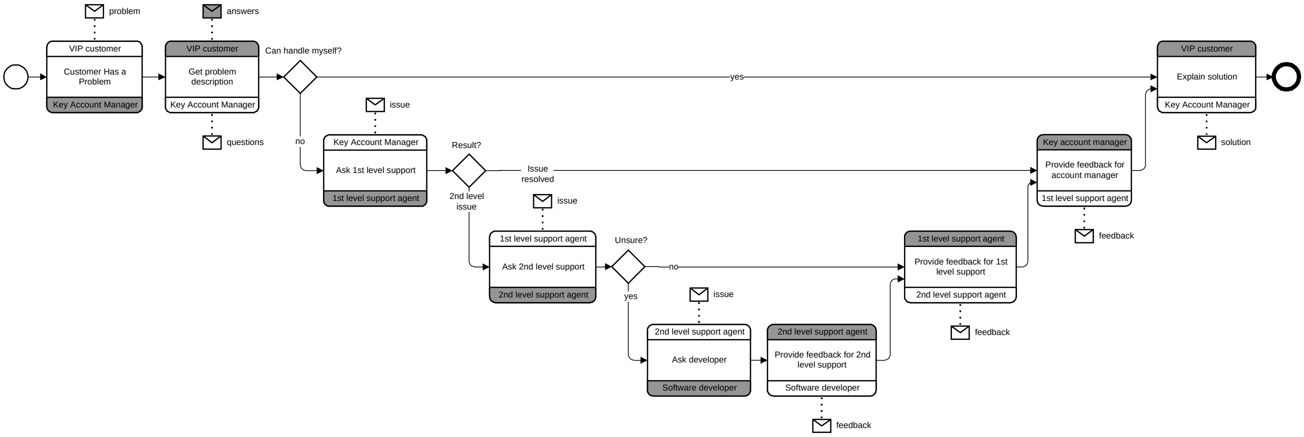

Figure 6.3: Incident Management as choreography

6.3 Human-driven vs. system-driven control flows

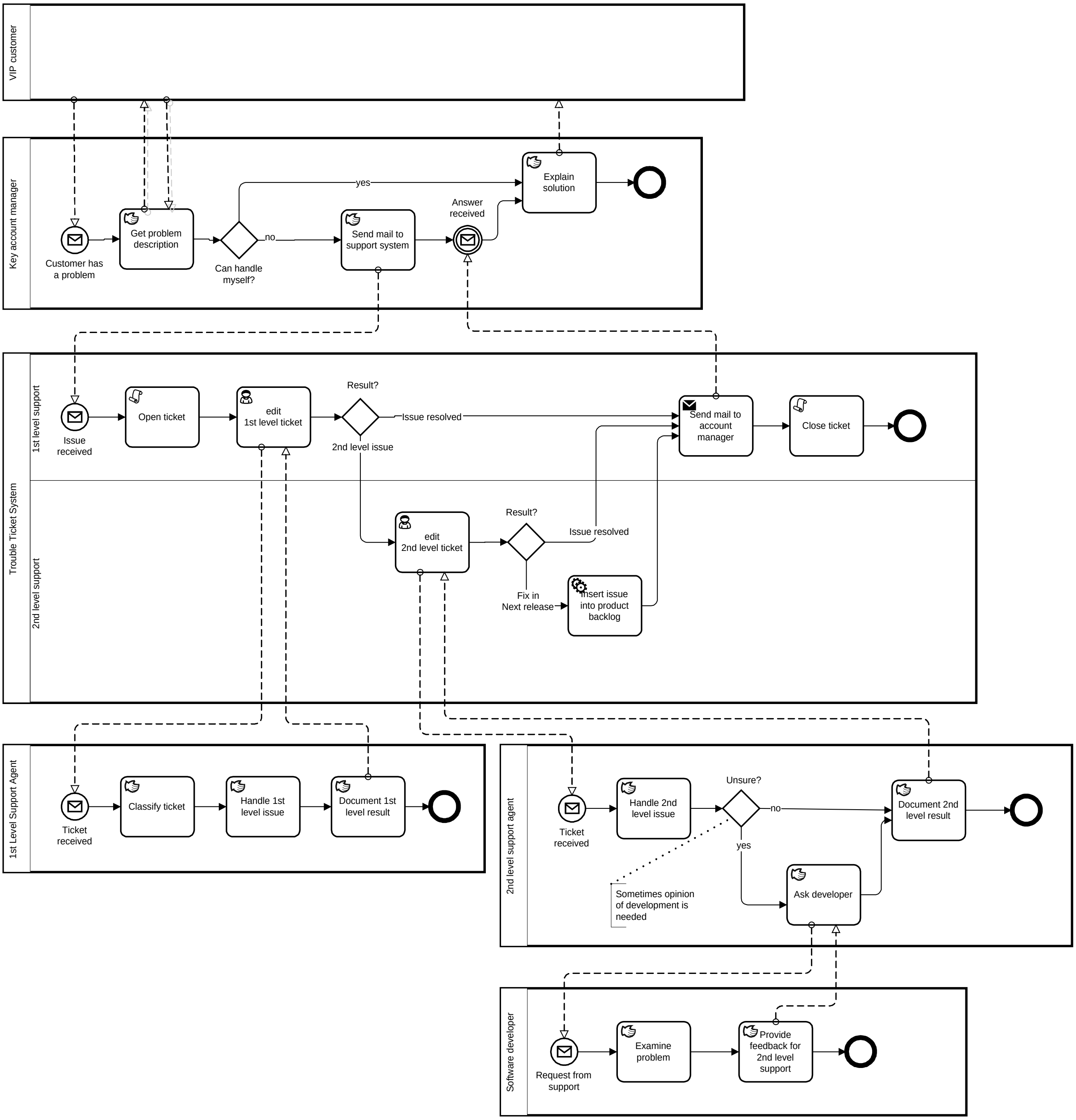

Figure 6.4: Incident Management with human-driven and system-driven pools

If we imagine we are realizing a project for automating the incident management process, we could now decide which parts of it should be actually executed in a process engine, and which parts should remain human-driven. In this scenario we decided that the account manager should not be bothered with web forms or task lists, he should just send an email if he wants to report a customer's problem, and receive an email when the process has completed. The same idea applies for the software developer: Let us assume the 2nd level support agent sits in the same room as the developers. Maybe it is more efficient if the support agent just walks over to the developer and talks about the issue, rather than playing some time consuming ping-pong-game with task assignments. Therefore, we want to keep this part of the incident management human driven as well: no process engine driving the collaboration between 2nd level support and software developers. But