

For the programmer building a game or virtual world in the Project Darkstar environment, the visible architecture is the set of services contained in the stack. The overall set of services is both changeable and configurable, but four basic services will always be present and form the core of the operating environment, as shown in Figure 3-2.

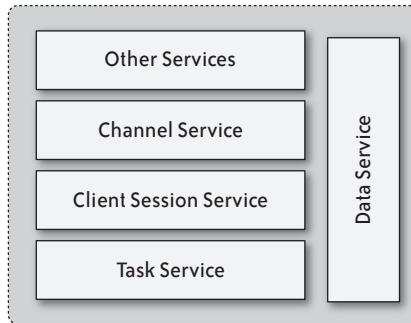


FIGURE 3-2. *Darkstar stack*

The Basic Services

The most basic of these stack-level services is the Data Service, which is used to store, retrieve, and manipulate all persistent data in the game or virtual world. The notion of persistence here is somewhat broader than might be found in other systems. In games or virtual worlds written in the Project Darkstar environment, any data that lasts longer than a single task is considered persistent and must be stored in the Data Service. Remember that we assume (and require) a programming model in which tasks are short-lived, so almost all of the data used to represent the server-side representation of the game or world will be persistent. The Data Service also knits together the separate copies of the game or world that are running on different servers, as all of these copies will share a single (conceptual) instance of the Data Service. All of the copies will have access to the same data, and all of the copies can read or change data stored in that service as needed.

Although the Data Store looks like a natural place for using a database, the requirements on the store are in fact very different from those that usually condition standard databases. There are very few static relations between the objects in the store, and there is no requirement within the game for any type of complex queries over the contents of the store. Instead, a simple naming scheme suffices, along with program-language-level references to the objects. The Data Store also has to be optimized for latency rather than throughput. The number of objects accessed by any particular task tends to be small (our preliminary measurements based on some prototype games and worlds suggest about a dozen objects per task), and about half of those objects that are accessed by any task are altered in the course of the task.