

Architecture Of An AtScale Cluster

A Clustered AtScale installation consists of an external load balancer and three hosts: two application hosts and one service coordinator host.

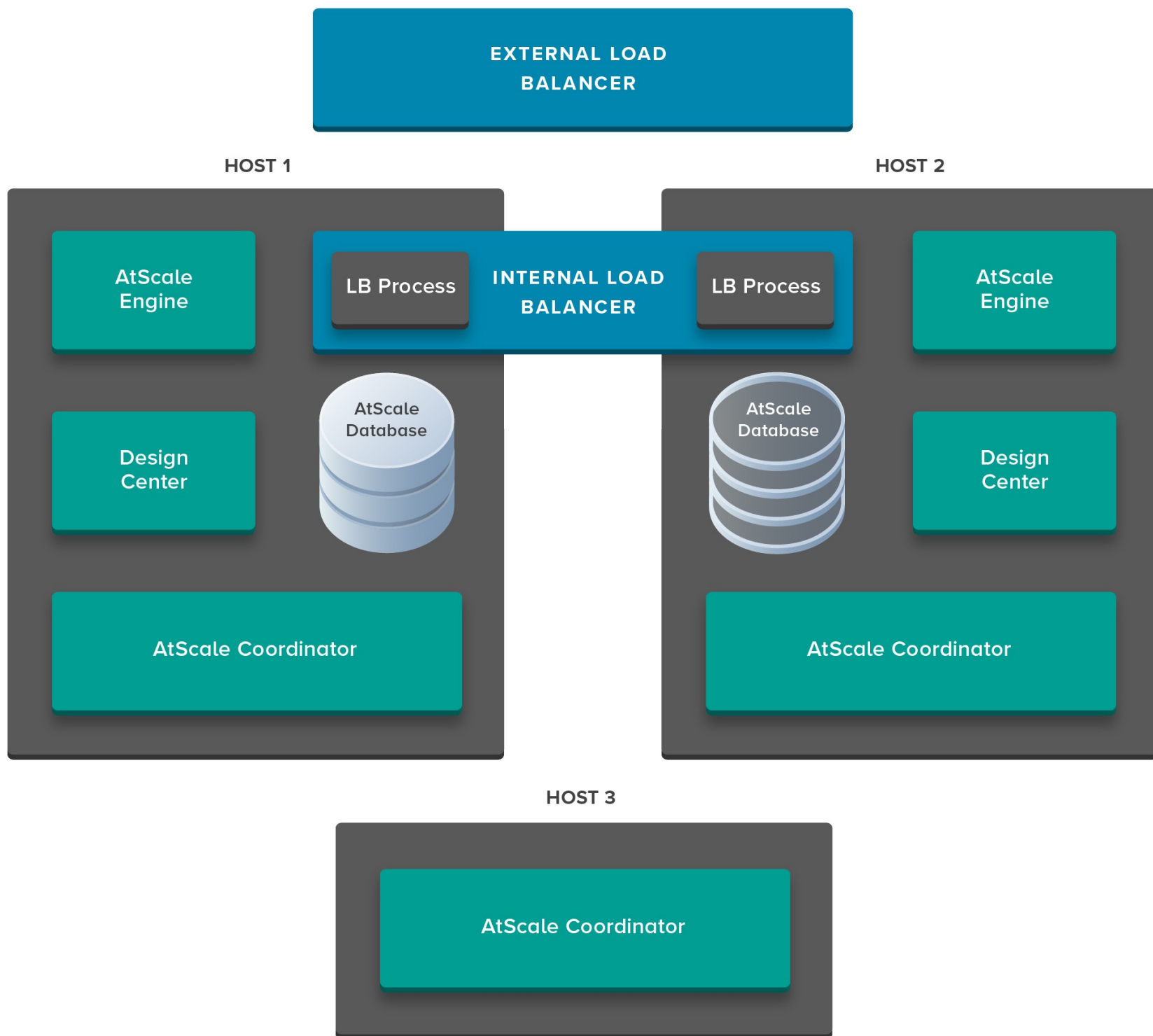
Two hosts in the cluster run the core AtScale services:

- ▲ An instance of the AtScale engine
- ▲ An instance of the Design Center application, which manages all services
- ▲ An instance of the AtScale database (Postgres)

One host runs only the AtScale Coordinator.

Each instance also runs an internal load balancer, which handles communication from the AtScale applications to the master AtScale database instance running on one of the hosts.

Figure 1. Components of a Clustered AtScale installation



Note: The figures here show the main components of a Clustered AtScale installation. The host identifiers 1 through 3 and service locations depict a hypothetical installation. This document does not intend to dictate host names or to pin certain services to specific hosts.

Host 1

The instance of the AtScale engine is running and active on this host.

The Design Center application is initially running and active on this host. If this application fails, the instance of the application on the other host becomes active and takes over.

The instance of the AtScale database on this host is initially the master AtScale database instance for the cluster. The master was either started up first, or was intentionally converted to the master through a manual or automatic failover action. If the master fails, the database instance on the other host becomes active and takes over from the failed database instance.

Host 2

The AtScale engine, the internal load balancer, and the Design Center application are running and active on this

host. The AtScale database on this host is on warm standby.

- ▶ Although the AtScale database instance is not being used, is running and replicating the changes made to the master database.

Host 3

The AtScale Coordinator service runs on this host. This service is required to maintain the cluster state and to initiate fail-over actions. This service does not process any business intelligence workloads.

External load balancer

You must configure a load balancer to handle queries from client business intelligence (BI) applications such as Tableau Desktop and Microsoft Excel, requests from browsers which are connected to the Design Center, and communication between AtScale applications. Your load balancer can be a software (for example, HAProxy) or a hardware (for example, F5 or Cisco) appliance. AtScale provides a script, `generate_ha_proxy_cfg`, for you to generate an example HAProxy 1.7 load balancer configuration.

Internal load balancer

An AtScale cluster is automatically installed with an internal load balancer on the hosts 1 and 2. The internal load balancers do not communicate directly with each other, but provide access to the master AtScale database.

AtScale Coordinator

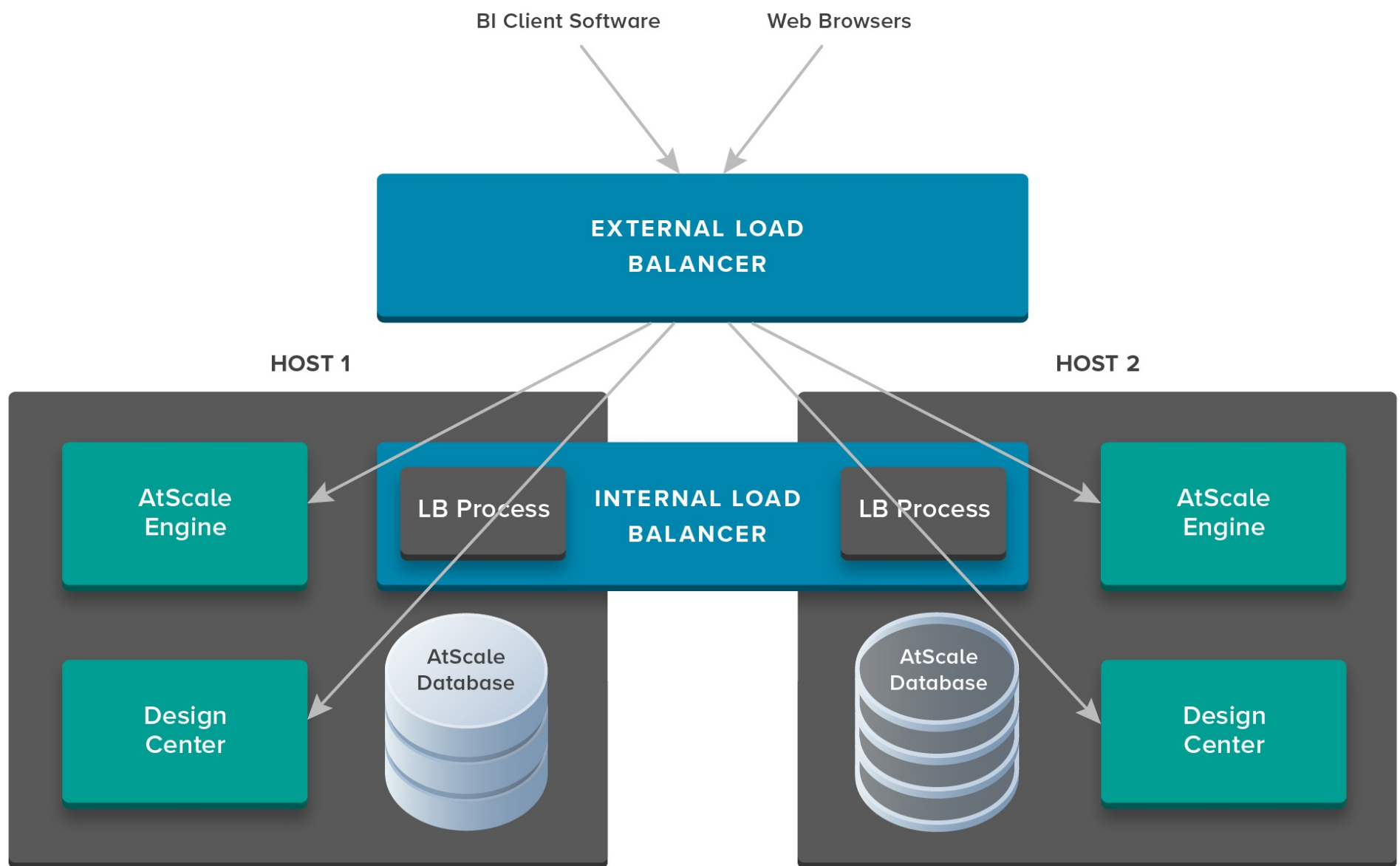
The AtScale Coordinator service maintains the state of the cluster for the AtScale engine and AtScale database services. The AtScale Coordinator hosts must be able to reliably communicate with each other. Also, each AtScale engine and AtScale database must have reliable network access to the AtScale Coordinator service and to each other.

How Client BI Software And Browsers Communicate With An AtScale Cluster

The external load balancer is the intermediary between the AtScale cluster and BI client software or browsers through which users are connected to the Design Center.

Requests and queries are passed on by the external load balancer to the appropriate AtScale application. The AtScale engine and the Design Center application on each host are both active and can each accept queries and requests. The external load balancer, which you can configure with the aid of the output of a script that AtScale provides in its installer package, keeps track of which applications are active.

Figure 2. A load balancer distributing Requests

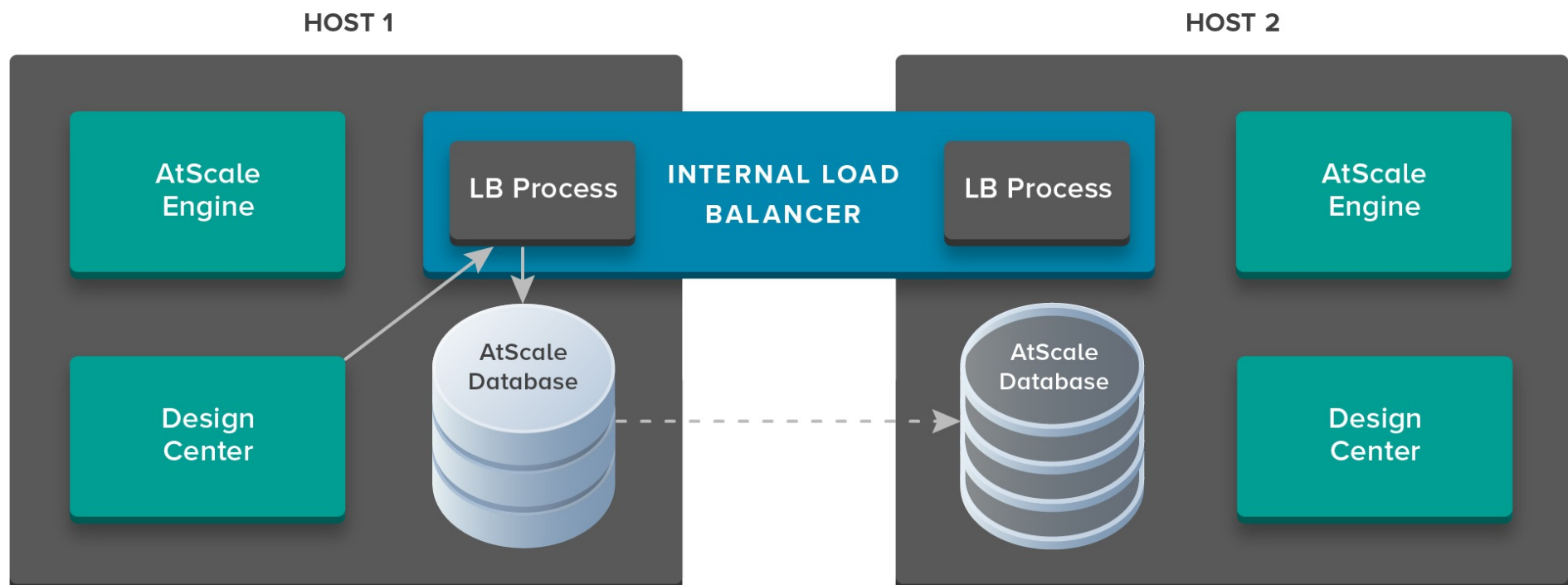


How Applications Communicate With The AtScale Database

While you are modeling data, managing users and permissions, and performing other actions, the Design Center application is sending updates to the master instance of the AtScale database. Therefore, if Design Center fails over to the other Design Center application while you are, for example, modeling data in a cube, the cluster automatically fails over and retrieves information from the master instance.

Figure 3 shows the message routing from the active Design Center to the master instance of the AtScale database.

Figure 3. The Design Center updates the database

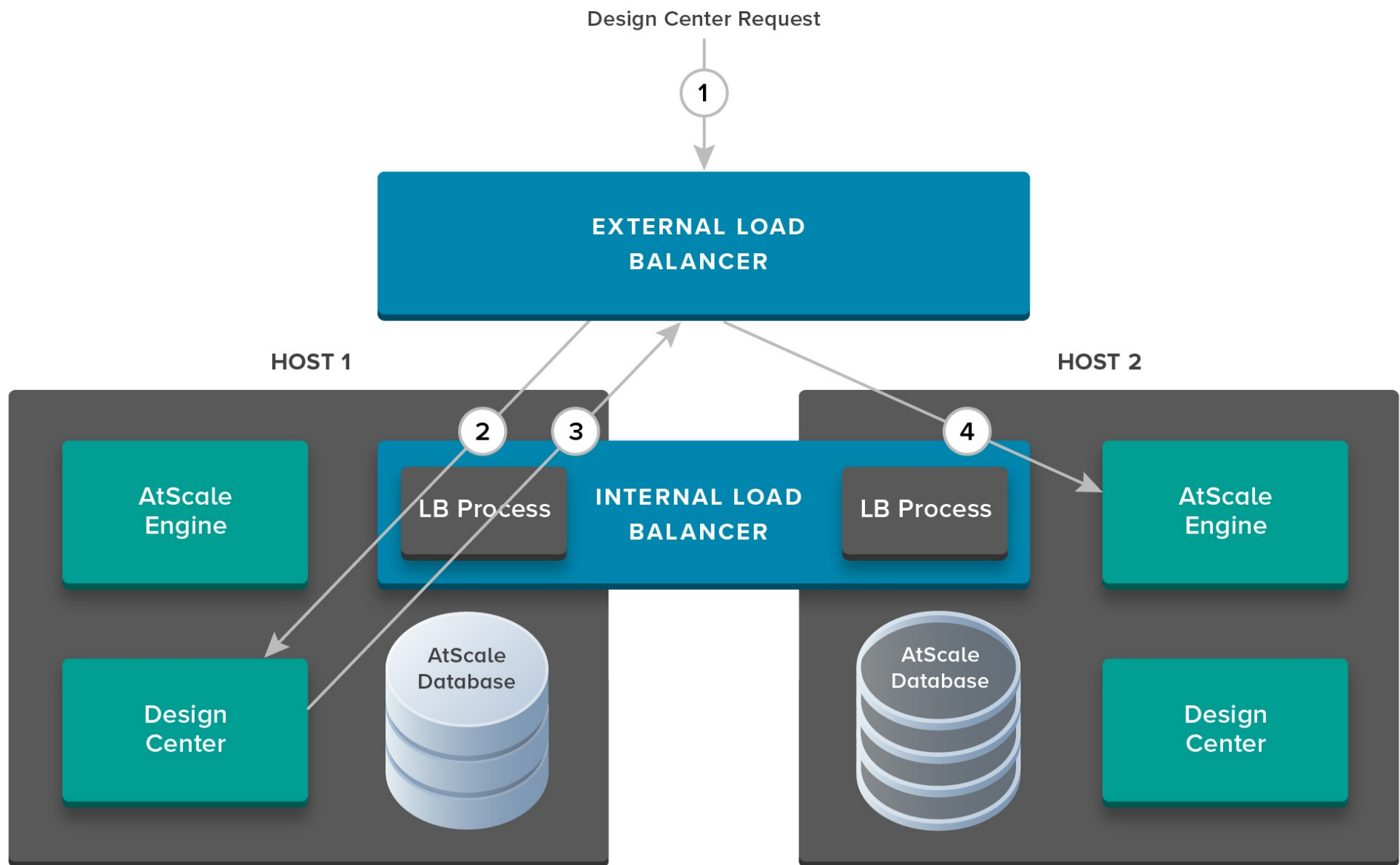


AtScale applications don't communicate directly with the database. They don't keep track of which database instance is currently the master instance. Instead, the internal load balancers keep track. An application that wants to update or retrieve data from the database sends its request to an internal load balancer, which then routes the message to the master database instance. If the request is an update, insert, or delete, the master instance replicates the change to the standby instance.

How Applications Communicate With Other Applications

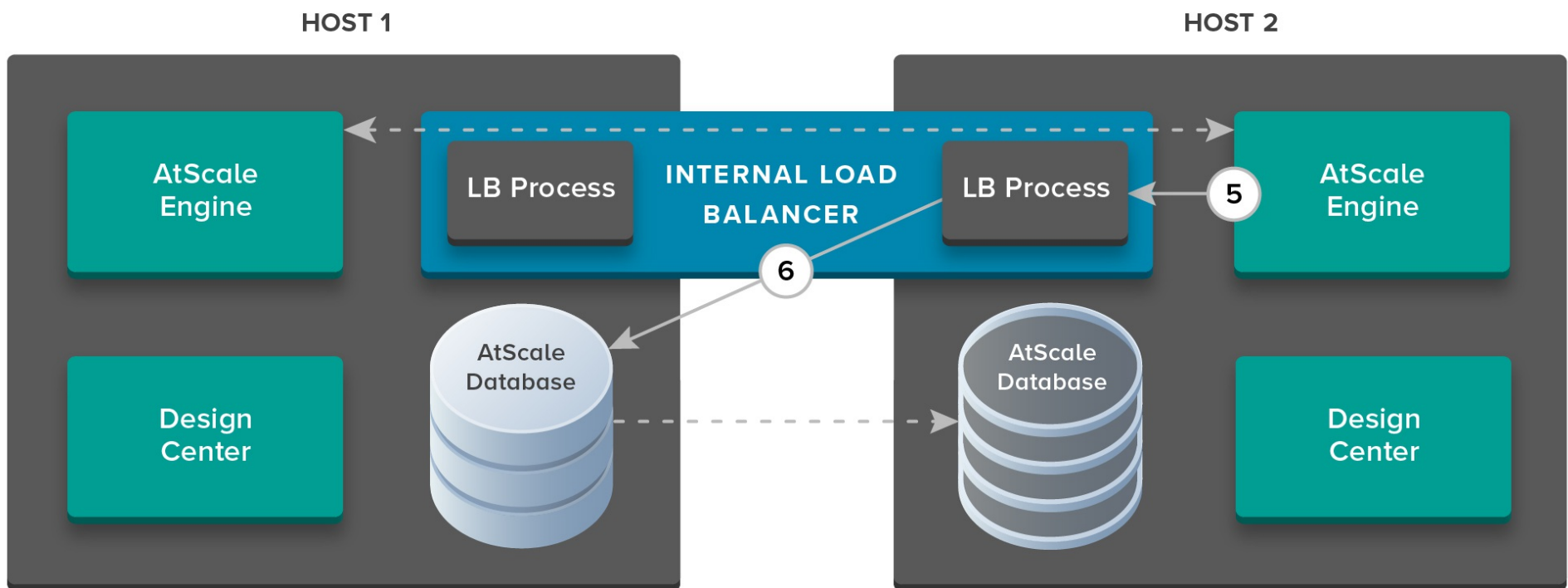
Actions that you take in AtScale can require applications to communicate with each other. For example, the Design Center communicates with the AtScale engine to authorize and coordinate actions such as viewing table data and publishing projects. The following diagrams show how this coordination takes place.

Figure 4. The Design Center sends a request to the engine



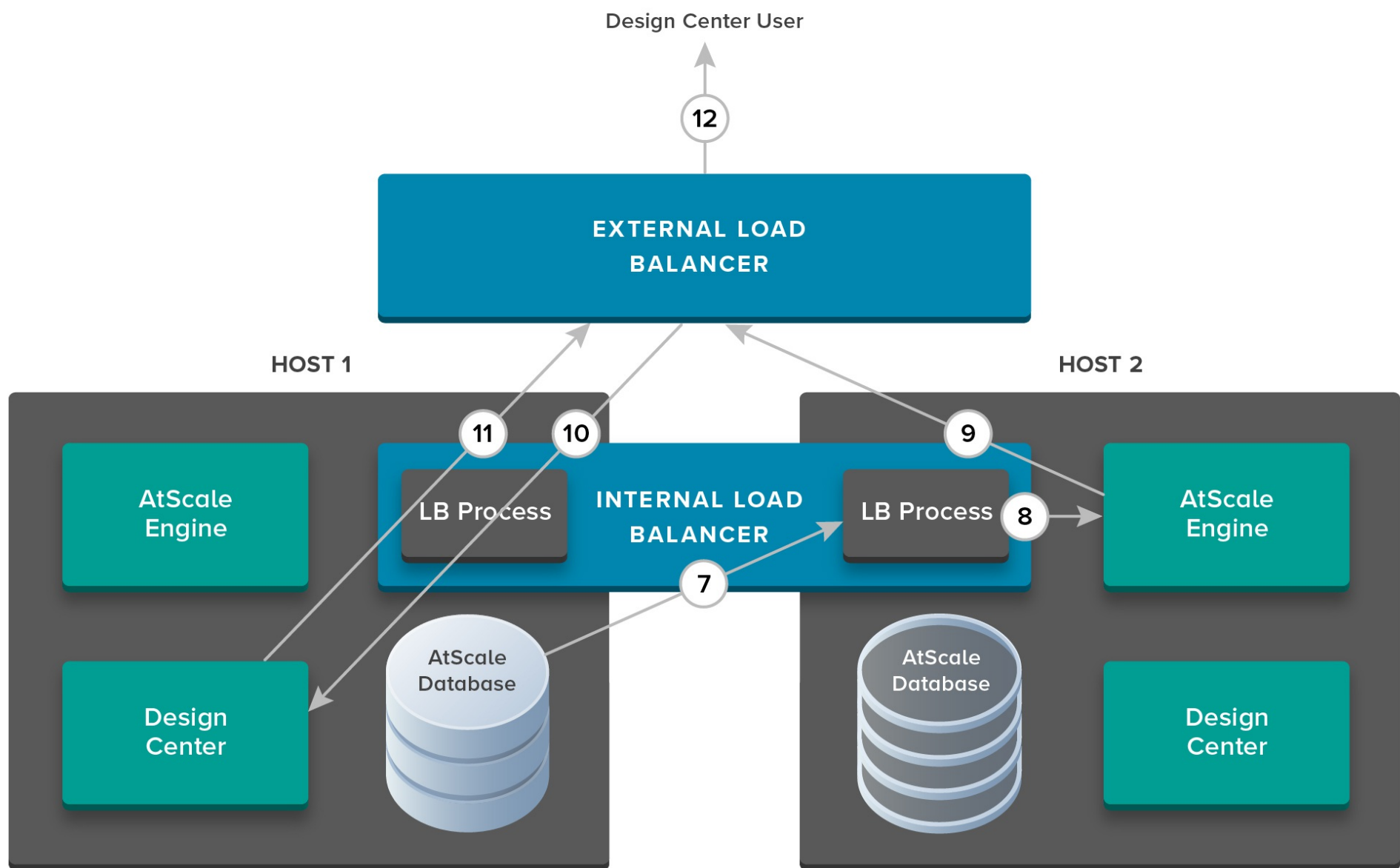
1. A Design Center request from a client is received by the external load balancer.
2. The load balancer passes the request to the Design Center.
3. The Design Center sends a message to the AtScale engine. The Design Center does not keep track of whether both AtScale engine instances are active or only one is active, but the external load balancer does. Therefore, the Design Center directs its message to the load balancer.
4. The external load balancer decides which engine instance to send the message to.

Figure 5. The engine updates the AtScale database instance



5. Because both instances of the AtScale engine are active, the engine instance that received the message from the Design Center sends to the other instance any information needed to keep the two instances identical.
6. The engine's information update goes through the internal load balancer.

Figure 6. The engine communicates back to the Design Center



- 7. and 8. The updated instance sends a message back to the AtScale engine through the internal load balancer.
- 9. and 10. Although the engines communicate directly with each other, they communicate with other applications through the external load balancer. So, the engine instance sends a message back to the Design Center as part of coordinating the action, routing the message to the external load balancer.
- 11. and 12. The Design Center sends a message back to the client through the external load balancer.