

# CloudStore Description

CloudStore describes a legacy bookshop to be migrated to cloud computing environments. Figure 1 gives a simplified overview of a Palladio model we created for CloudStore. To understand the model, a short description follows.

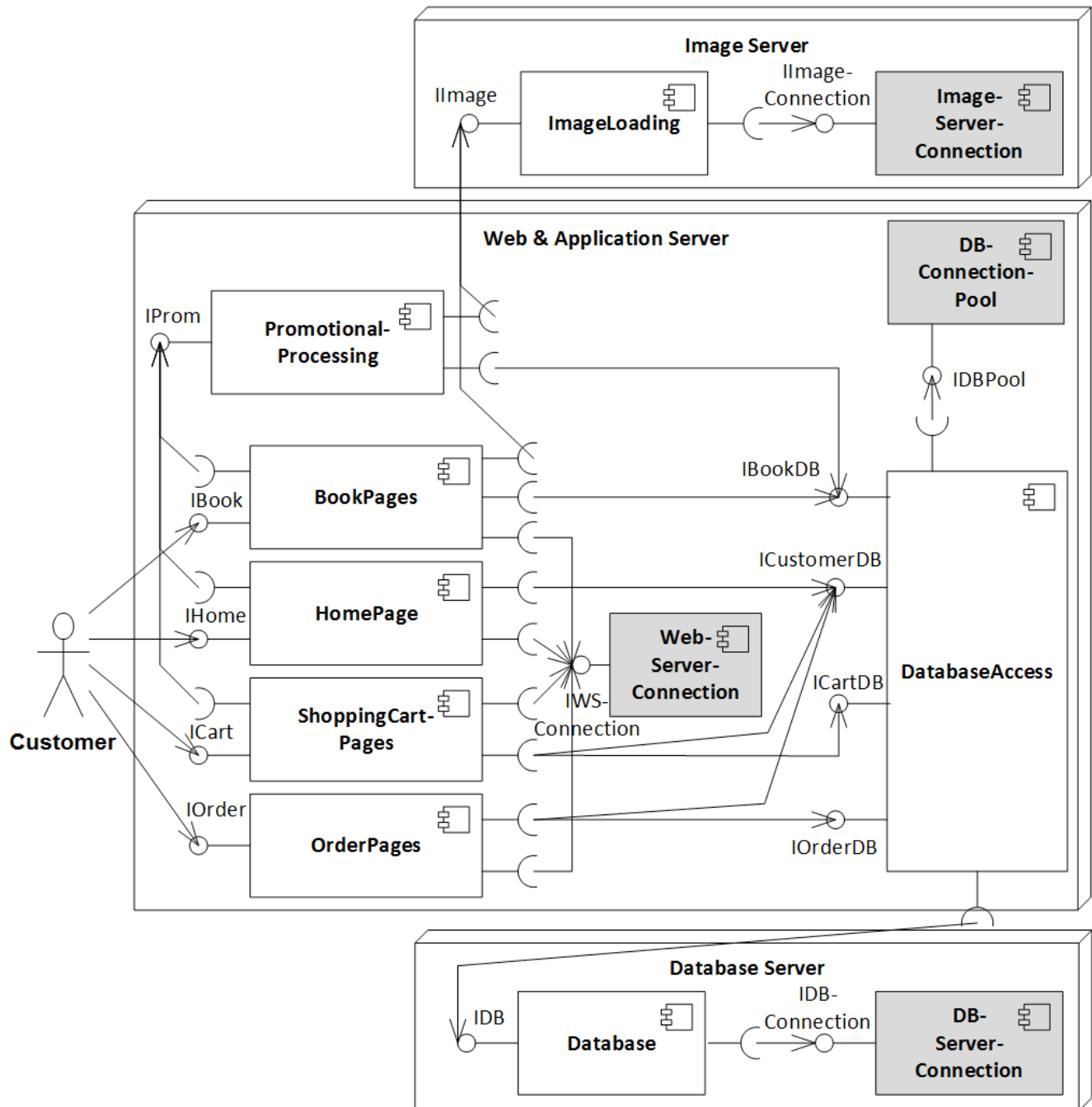


Figure 1: System Overview of CloudStore (Simplified)

Customers enter the CloudStore system via the web pages provided by front-end components. These front-end components are **BookPages**, **HomePage**, **ShoppingCartPages**, and **OrderPages** allocated on the **Web & Application Server**. **BookPages** provides operations regarding books (e.g., to query book details or to search for books). The **HomePage** component shows CloudStore's home page, which welcomes its customers and displays possible book categories for browsing. **ShoppingCartPages** allows customers to register, add books to a shopping cart, and to check-out the shopping cart. Afterwards, **OrderPages** allows to follow-up on the order. **BookPages**, **HomePage**, and **ShoppingCartPages** additionally require the **PromotionalProcessing** component to receive an advertisement area for related books.

These front-end components require operations of the **ImageLoading** and **Database** components as allocated on the **Image** and **Database Server**, respectively. **ImageLoading** provides access to image files, e.g., needed for book covers. CloudStore's **Database** stores entries for books, customers, shopping carts, and orders.

Calls to the **Database** are intercepted by the **DatabaseAccess** component that manages database connections. **DatabaseAccess** receives [returns] such connections from [to] the **DBConnectionPool** component. Also **Web & Application**, **Image**, and **Database Server** use pools for handling customer requests (**WebServerConnection**, **ImageServerConnection**, **DBServer-Connection**). All of these pools (gray-colored components in Fig. 1) are typical factors that influence an application's performance as their pool-size limits the amount of requests that can be processed in parallel.

Palladio supports acquiring and releasing connections from these resource pools in service effect specifications (SEFFs). SEFFs specify the behavior (control and data flow) of component operations. In our model, every interaction requires the acquisition of connections and a subsequent release once the interaction ends. Figure 2 illustrates this pattern for SEFFs of front-end component operations that interact with database and image components. Actions (1) to (3) model the performance impact of creating an HTML page for customers while action (4) models the performance impact of subsequently resolving image references. These two phases—receiving an HTML page and subsequently its references—reflect the typical behavior of web browsers.

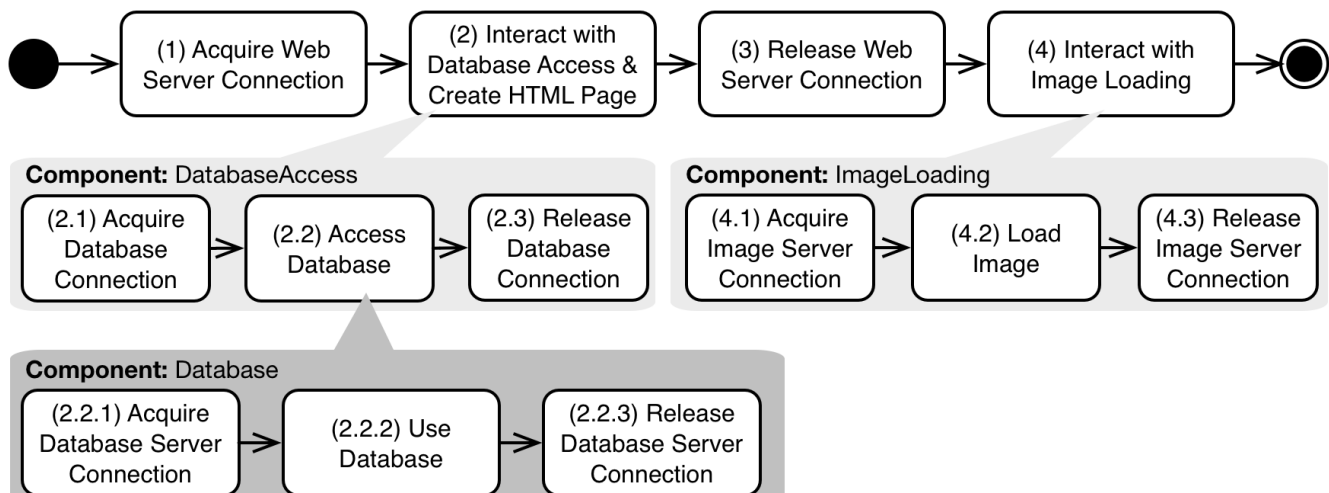
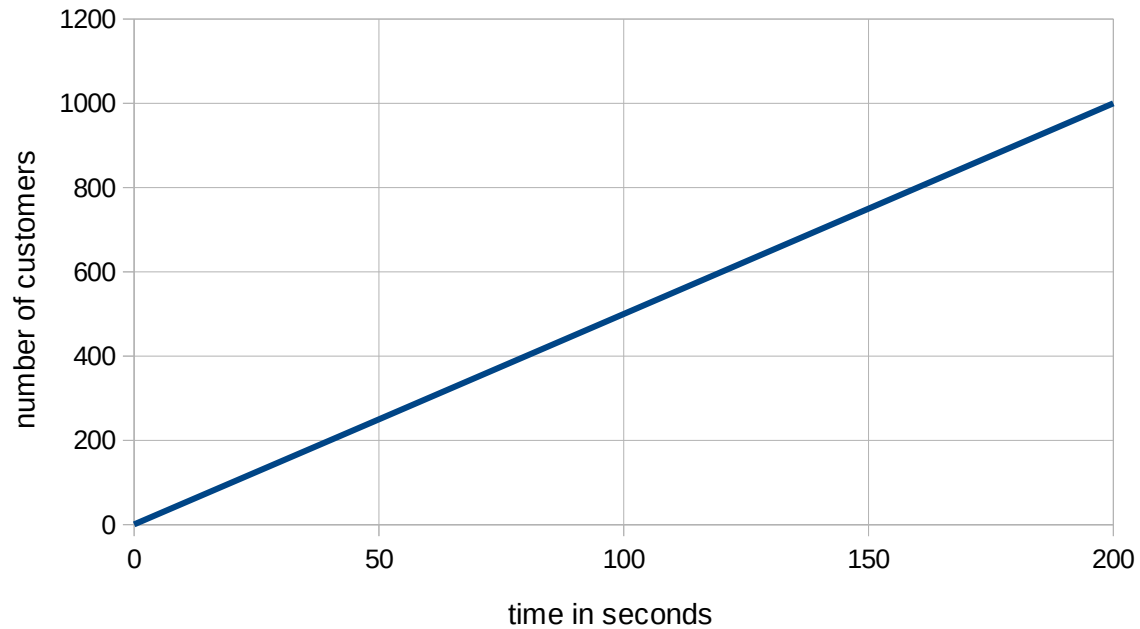


Figure 2: Behavior of front-end components interacting with database and image components

In Figure 3, the load to CloudStore is illustrated. At the x-axis, the simulation time is depicted. At the y-axis, the number of concurrent customers within the system is depicted. Accordingly, at simulation start, there is only one customer within the system. Afterwards, the number of customers linearly increases over time. At the end of the simulation time (200 seconds), there are 1000 customers within the system. The distribution of the customers within CloudStore is defined in the specified usagemodel (UsageScenarios/controlledExperiment.usagemodel).



*Figure 3: Number of customer over time*