3 Types of Integration

There are many different ways to integrate enterprises with each other, with their various autonomous divisions and with their hosted or nonhosted back-end application systems. In order to derive the spectrum of integration use cases for a generic B2B integration technology architecture it is mandatory to explore and to examine integration scenarios in more detail and in more depth. This chapter reviews an exhaustive set of integration scenarios from an integration technology point of view. The scenarios are described graphically.

This chapter shows the spectrum of integration use cases that a B2B integration technology must be able to solve. The variety of potential integration situations is introduced. Chapter 4 provides a classification of the integration functionality itself in order to classify different integration technologies independent of integration use cases. This allows us to characterize different integration technologies by their functionality. Both this chapter and Chap. 4 provide a frame of reference for concrete integration situations as well as specific integration technology. The integration scenarios are described as well as depicted graphically. Figure 3.1 shows the legend of the basic set of graphical symbols used in the following sections.

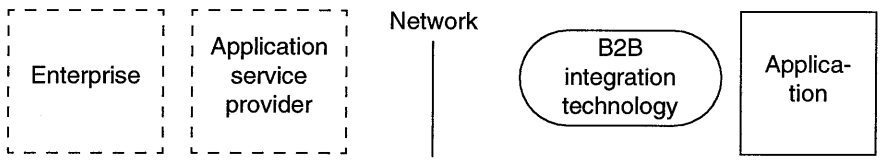

Fig. 3.1. Legend of symbols for integration scenarios

The symbol for enterprise depicts the integration boundaries of an enterprise. As is later shown, back-end application systems, B2B integration technology servers and other components are inside the enterprise icon since they are part of the internal software components of the enterprise. The symbol for application represents any type of back-end application system, including legacy systems as well as "out-of-the-box" application systems, that is, systems that are bought instead of built by the enterprise. The symbol for the B2B integration technology server represents the B2B integration technology server used for the specific integration scenarios. The symbol for networks is a line since the widely used cloud representation takes too much space in figures. Where necessary, the graphical symbols are annotated in order to clarify a scenario.

An important note is that whenever two or more B2B integration technology servers are shown inside or outside an enterprise or application service provider, they could be from different vendors, i.e., they are not necessarily the same technology or product. This note is important since the graphical notation does not imply that the same technology from the same vendor has to be used.

The integration scenarios assume that the B2B integration technology provides the functionality required for the scenarios. Any particular type of integration technology as discussed in Chap. 1 could be used in reality, as long as it provides the necessary functionality. Of course, mature B2B integration as specified in Chap. 2 would be fine, too.

3.1 Point-to-Point Back-end Application System Integration

The simplest integration scenario is point-to-point integration between two back-end application systems within one enterprise. This scenario represents the case in which two applications exchange data with each other (Fig. 3.2).

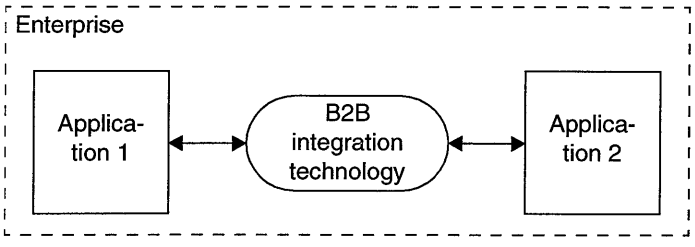

Fig. 3.2. Point-to-point integration

Each of the two back-end application systems is connected with the B2B integration technology server. The B2B integration technology server provides the functionality discussed in Chap. 1 or Chap. 2 in order to connect the two back-end application systems.

3.2 Multipoint Back-end Application System Integration

Multipoint integration refers to the case where three or more back-end application systems are communicating with each other. Figure 3.3 shows the topology with three back-end application systems.

It is not necessarily the case that each back-end application system communicates with every other one. Multipoint integration can fundamentally be of two

types. In the first type, each pair of back-end application systems is communicating in a point-to-point relationship (or not at all, if two back-end application systems are not communicating at all). This means that point-to-point functionality as discussed in Sect. 1.2.2 is sufficient.

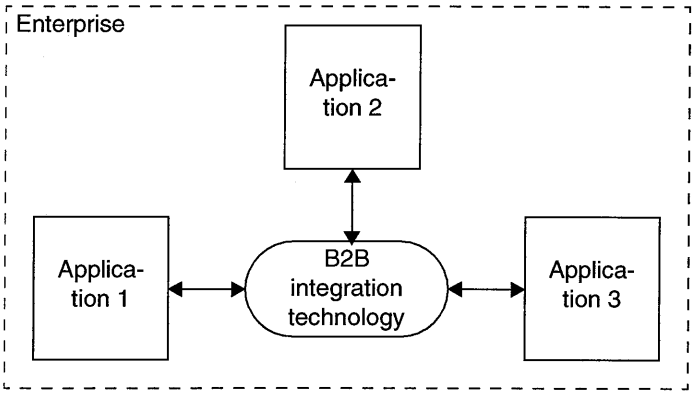

Fig. 3.3. Multi-point integration

In the second type, in addition to pairwise communication three or more back-end application systems are exchanging messages through a business process. For this case business process management functionality as discussed in Sect. 1.2.4 is required.

3.3 Back-end Application System Metadata Synchronization

The "smallest" integration scenario is when a back-end application system communicates with the B2B integration technology server itself to exchange data like trading partner data. This scenario requires some more detailed explanation. Trading partner data contain information about a trading partner's name, unique identification as well as network addresses for communication. back-end application systems manage trading partner data for their purposes so that they can refer to trading partners internally. For example, an Enterprise Resource Planning (ERP) system stores purchase orders, and each purchase order is related to the trading partner that either sent the purchase order or to which the purchase order was sent.

A B2B integration technology server has to do the same for providing its functionality. Each message it processes comes from a trading partner or is sent to a trading partner (in the business-to-business case). In order for a B2B integration technology server to be able to identify trading partners, it uses the unique identifiers that are stored within its trading partner management subsystem.

When a back-end application system sends business data to the B2B integration technology server, it refers to a trading partner's unique identifier indicating the

target trading partner of the message. The B2B integration technology server has to send the message to the trading partner with that unique identifier. In order for the B2B integration technology server to interpret the unique trading partner identifier the same way (i.e., refer to the same real trading partner), the trading partner data must be synchronized between the back-end application system and the B2B integration technology server. For example, if a trading partner is added to the list of trading partners in the back-end application system, then the same trading partner needs to be added to the B2B integration technology server's trading partner subcomponent. In order to accomplish the synchronization of trading partner data, the back-end application system needs to be integrated with the B2B integration technology itself. Figure 3.4 shows the topology with one back-end system application. Chapter 9 discusses in more detail how the trading partner synchronization can be accomplished using the integration concepts provided by the B2B integration technology server themselves.

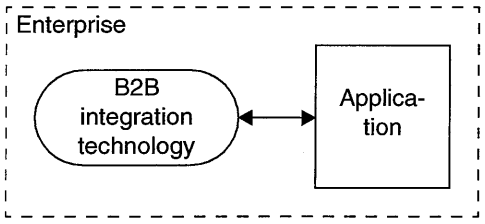

Fig. 3.4. Metadata synchronization

This type of communication is called metadata synchronization since the integration is about data that is necessary to accomplish the regular business data integration, like purchase orders. The messages exchanged contain data about trading partners like their unique identifier or business phone number, not business data like a shipment notice. If more than one back-end application system is connected (the usual case to be expected) then in general each has to be included in trading partner synchronization.

3.4 Supply Chain Integration of Two Trading Partners

This case can be illustrated by the almost "classical" example of two trading partners exchanging a purchase order and a purchase order acknowledgment. Fundamentally, two trading partners exchange business data over a network in order to establish a trading relationship, to exchange business data, notifications or status messages. Each trading partner uses a B2B integration technology server to execute the communication (Fig. 3.5).

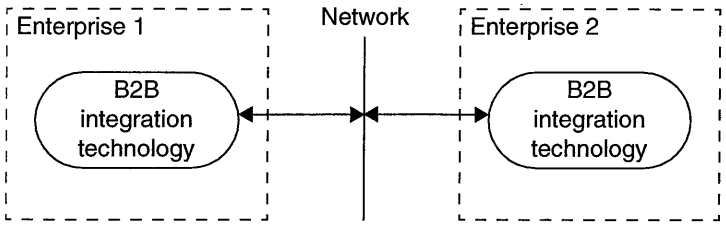

Fig. 3.5. Supply chain integration of two trading partners

Usually, the trading partners store or retrieve the business data from back-end application systems. Therefore the B2B integration technology deployed is integrated with the back-end application systems used by the enterprises. Figure 3.6 shows a more complete supply chain topology.

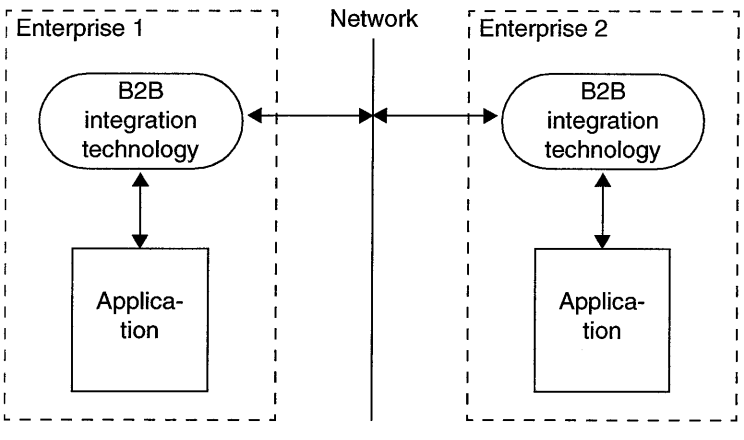

Fig. 3.6. Supply chain integration of two trading partners and their back-end application systems

In the general case each trading partner has different back-end application systems as well as a different number of systems. Also, each trading partner might have B2B integration technology from a different vendor so that both trading partners cannot rely on having any technology in common at all. This requires true interoperability on a B2B protocol level.

3.5 Supply Chain Integration of Several Trading Partners

As soon as more than two trading partners are involved in a supply chain, then it is likely that not every trading partner involved is communicating with every other any more. For example, a transport company communicating with a supplier does

not necessarily communicate with the seller. Figure 3.7 shows the communication topology. The back-end application systems of the trading partners are omitted in the graphical representation. Since between each pair of trading partners there can be a different type of network (e.g., Internet or VAN), a separate network is depicted.

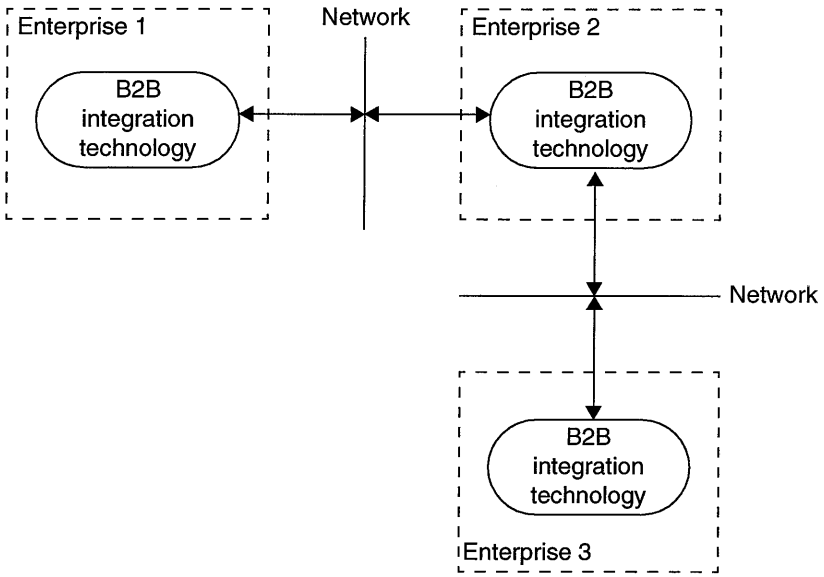

Fig. 3.7. Supply chain integration of several trading partners

Each trading partner involved in this example has bilateral communication relationships, but not every trading partner communicates with every other trading partner. Of course, it is possible that every trading partner in a supply chain is connected to every other trading partner to exchange business data. The B2B integration technology server must be able to deal with all cases.

3.6 Remote Back-end Application System

Small trading partners do not necessarily want to buy, install and maintain a B2B integration technology server in addition to the back-end application systems they already have. In case they have a small number of back-end application systems that do not communicate with each other, they might want to connect to their supply chain directly without running a B2B integration technology server. In this case the back-end application systems need to be "B2B integration enabled."

One way to enable a back-end application system to connect to trading partners is a dedicated B2B protocol back-end application system adapter that implements a particular B2B protocol, but cannot provide any other connectivity (for example, with other back-end application systems). In this scenario the back-end application system can directly connect to trading partners' B2B integration technology servers (Fig. 3.8). From a B2B integration technology viewpoint, the dedicated B2B protocol back-end application system adapter appears like any other B2B integration technology because the adapter implements a specific B2B protocol like any other B2B integration technology server.

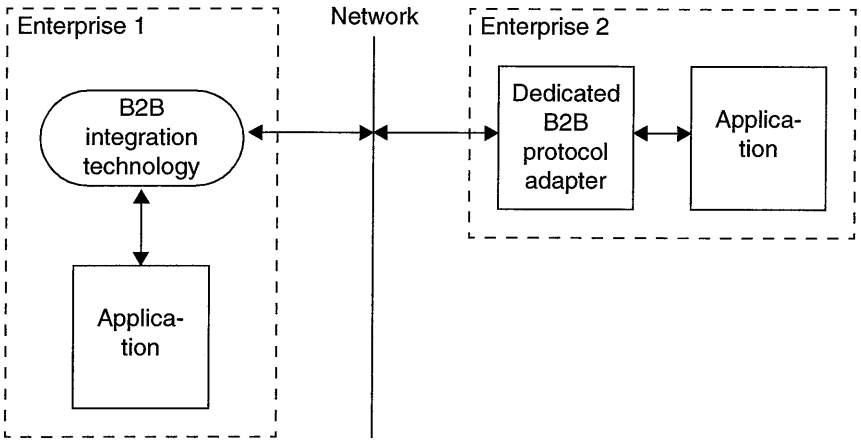

Fig. 3.8. Dedicated B2B protocol back-end application system adapter

3.7 Marketplace Integration

In its simplest form, a marketplace lists suppliers offering their products and allows buyers to access the supplier listings. Once a buyer finds a product they wish to buy, the marketplace has achieved its goal. All subsequent business data exchange, like a purchase order exchange, takes place outside the marketplace and independent of it (Fig. 3.9). A marketplace provides access to a seller and a buyer. Both are also connected directly for subsequent message exchange.

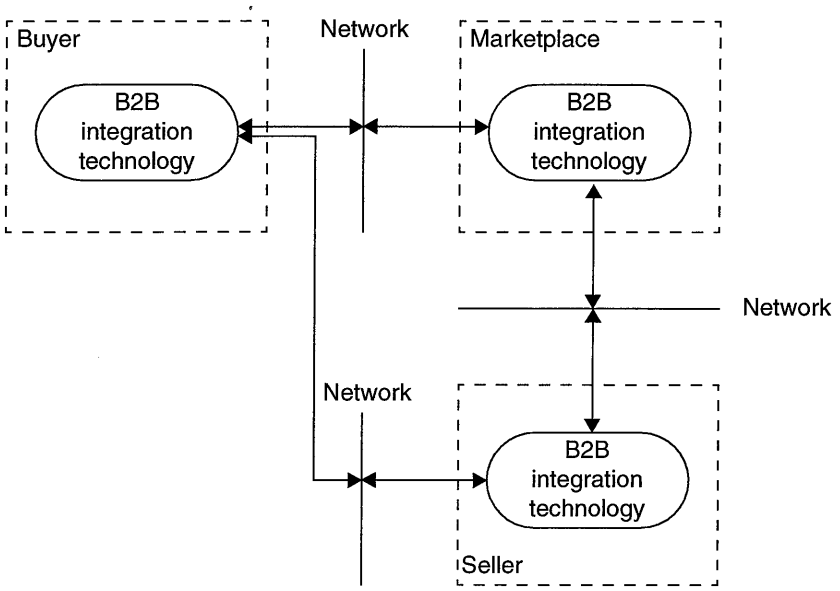

Fig. 3.9. Simple marketplace integration

More sophisticated forms of marketplaces provide automatic buyer/seller matching and might also issue the purchase orders and manage the purchase order acknowledgment as a response. In these cases the business data exchange is also managed by the marketplace. It might or might not be necessary for trading partners connected to a marketplace to exchange business data between themselves in addition to with the marketplace. In the extreme case, the marketplace provides all business data exchanges so that the trading partners do not have to connect with each other directly at all. In the latter case, no direct communication is set up between trading partners. For the example in Fig. 3.9, this means that there is no direct communication relationship between the buyer and the seller.

3.8 Hub

An enterprise with many trading partners in general faces the problem that it has to support many B2B protocols, since its trading partners might have chosen different B2B protocols for B2B integration. An implication of this is that the enterprise has to support all transformations necessary from its internal representation of business data to those of the various B2B protocols. For each business event like a purchase order, one transformation per B2B protocol has to be implemented. If a business

event can be sent as well as received, two transformations (one for each direction) are necessary. The total number of transformations is therefore, in the worst case, the number of business events times two (directions) times the number of B2B protocols supported.

A high number of transformations means a significant maintenance effort for an enterprise. Large enterprises might be able to afford the maintenance cost, but smaller enterprises certainly would appreciate not incurring the cost. This problem is addressed by transformation hubs.

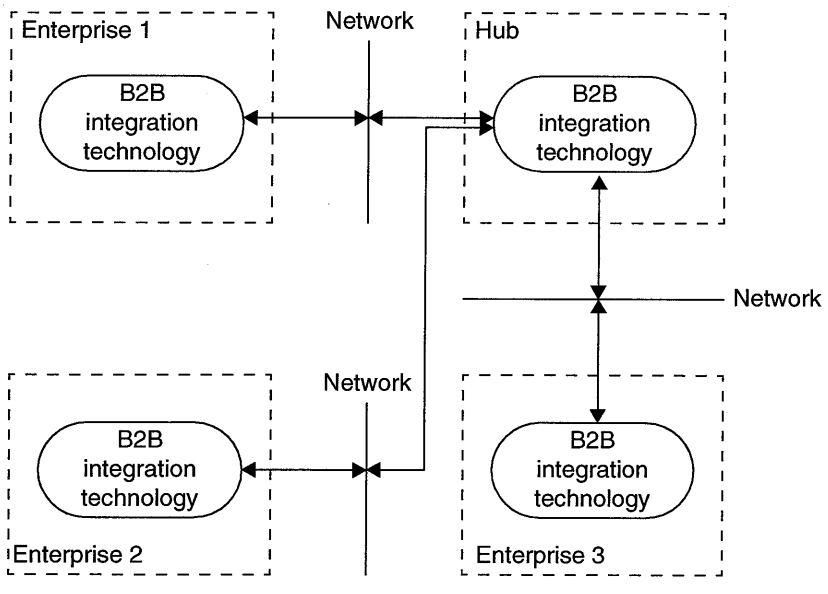

Fig. 3.10. Transformation hub

Transformation hubs implement the transformations between B2B protocols so that the enterprises do not have to do it themselves. Each enterprise communicates with the hub through one B2B protocol over a specific network. This requires that an enterprise only has to transform its internal business data to this one B2B protocol. The transformation hub takes care of implementing the high number of transformations (and most likely charges a fee for it). Once a hub receives a message and transforms it, it is forwarded to the final destination trading partner. Figure 3.10 shows the topology with one hub and three trading partners connected to it.

In addition to transformation, a hub can provide more services. For example, a hub can log the messages as they come into the hub and leave the hub for auditing purposes. At any time a trading partner can access the hub asking for all or a specific subset of messages that it has sent. Furthermore, the hub can provide analysis functionality that a trading partner can use. For example, the hub could analyze how many messages came in for a trading partner sorted by originating trading

partner. A hub can also provide time-stamping services so that messages are time stamped according to a single clock in the hub instead of the individual clocks in the trading partners. This might avoid issues in message processing when the clock times are significantly different.

3.9 Interactive Application

The discussion so far has not covered the case where an interactive application accesses the B2B integration technology server. In this case an interactive application like a browser or graphical user interface is integrated with the B2B integration server technology. A user using the interface enters the data as required by the user interface. Once finished with the entering of data upon submission, the interactive application connects with the B2B integration technology server and passes data on to it. An example is an employee procurement application, where employees select products to be purchased for internal use (like pencils). Once all products are chosen, the purchase order is issued through the B2B integration technology.

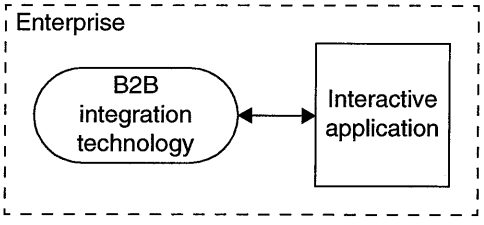

Fig. 3.11. Interactive application integration

Even though this sounds like a new type of scenario, in reality it is not. From the perspective of the B2B integration technology server, the interactive application is a back-end application system. The fact that a user drives the interaction through a user interface is not visible to the B2B integration technology server. The most common case is that the interactive application is inside the enterprise (Fig. 3.11).

The not-so-common case is that the interactive application is outside the enterprise boundaries. In this case there are two possibilities: either it connects to a B2B integration technology server that is also outside the enterprise boundaries (Fig. 3.12) or it connects directly across the network by complying to a B2B protocol that is supported by the B2B integration technology server (Fig. 3.13). In the first case, the two B2B integration technology servers exchange messages as if there were no interactive application. In the latter case, the interactive application is like a remote back-end application system, as discussed earlier.

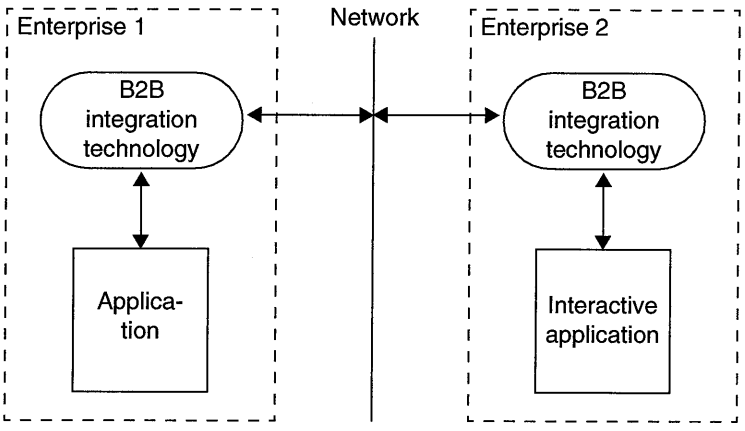

Fig. 3.12. Interactive application connected through remote B2B integration technology server

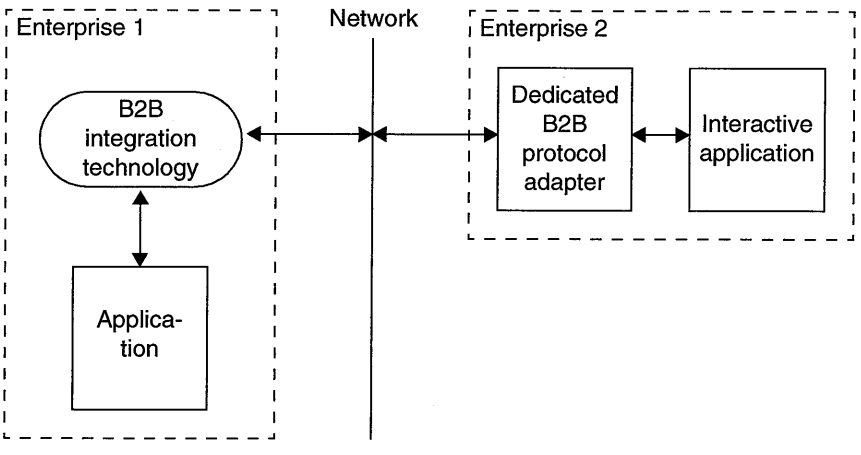

Fig. 3.13. Remote interactive application

3.10 Intra-Enterprise Integration

In addition to the scenario where enterprises connect with each other (like in the supply chain scenario), they can also connect various internal groups like divisions with each other. So far it is assumed that only back-end application systems are internal to enterprises and need to be integrated through a B2B integration technology server. However, larger enterprises consist of several divisions that are, in many cases, geographically distributed. These divisions can have their own set of

back-end application systems that they manage and that process division internal business data. In cases when they are comfortable providing access, one B2B integration technology server for the whole enterprise can manage all integration requirements. This server connects back-end application systems within divisions as well as across divisions.

Divisions in many cases feel uncomfortable providing access to their back-end application systems directly to the other divisions of the enterprise. In this case each division can deploy a B2B integration technology server by itself. Each B2B integration technology server integrates the applications of a division with each other. As soon as divisions need to exchange data between themselves, the B2B integration technology servers communicate with each other as if other divisions were external enterprises. Depending on the enterprise, they all follow the same B2B protocol, or different divisions can implement different B2B protocols. Figure 3.14 shows two divisions of one enterprise, each deploying its own B2B integration technology server.

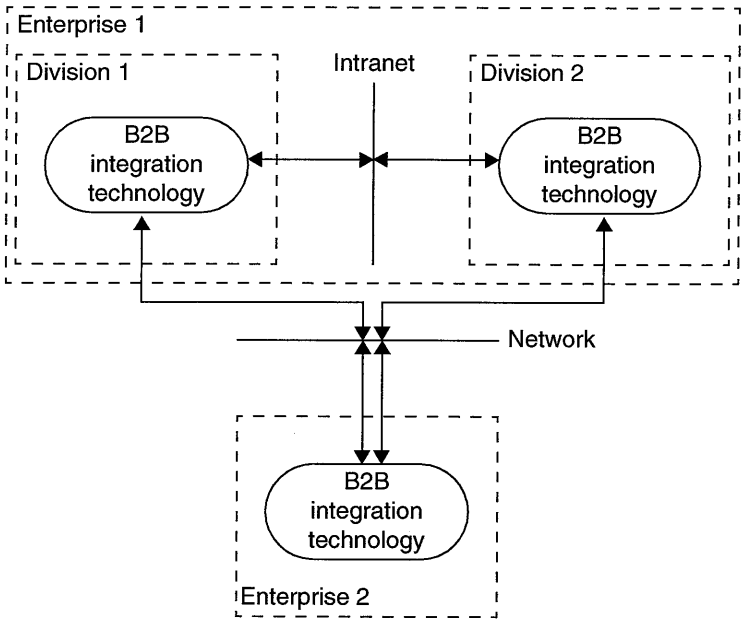

Fig. 3.14. Intra-enterprise integration with several B2B integration technology servers

Since each division has its own B2B integration technology server, each division can connect to external enterprises through B2B protocols individually. This means that there is no longer a single B2B integration technology server that facil-

itates interenterprise communication. It depends on the particular enterprise to set the policy on external communication. Figure 3.14 shows the case where every division connects to external enterprises individually.

If an enterprise decides to have only one B2B integration technology server connecting to external enterprises then it could be one out of those deployed in divisions. Alternatively, it could be a dedicated B2B integration technology server that does not belong to any particular division but to the enterprise as a whole instead. All divisions connect to this dedicated server in case they require connectivity to external enterprises.

3.11 Application Service Provider

As introduced in Chap. 1, application service providers (ASPs) install back-end application systems in their enterprise and make them available to their customers (hosted back-end application system). In general, customers access the user interfaces of hosted applications through browsers over the Internet.

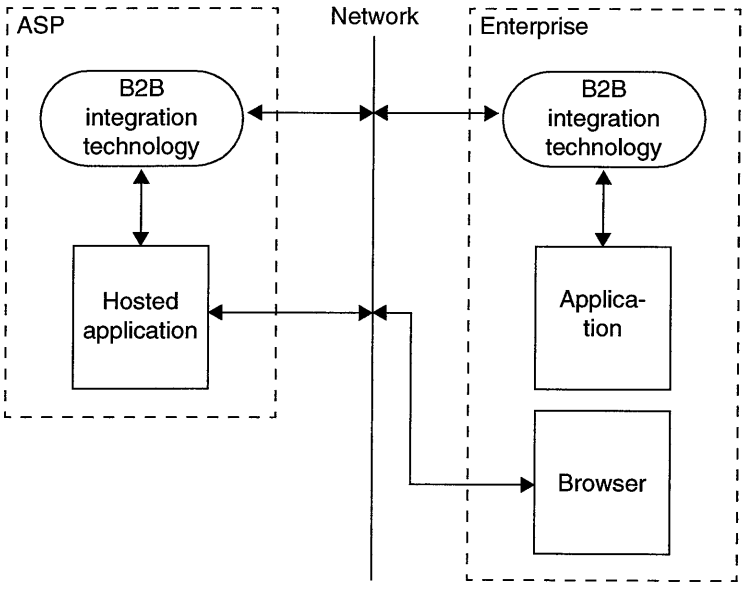

Fig. 3.15. ASP connectivity using a B2B integration technology server

However, in addition to accessing a hosted application through browsers, an enterprise wants to integrate the hosted application with its locally installed back-end application systems. In order to achieve this, it must be possible to access the hosted application systems as if it were locally installed. This is possible if an ASP provides access to the hosted application over a network through a B2B protocol.

Then it is possible that an enterprise can integrate its own hosted data with its own local data. Figure 3.15 shows the topology involving a B2B integration technology server. If an ASP hosts only one back-end application system, a dedicated B2B protocol adapter as discussed earlier might be sufficient to achieve the necessary connectivity with its customers.

An interesting "twist" is the following. If an enterprise 1 exchanges business data with enterprise 2 using a B2B protocol, then it can be that the enterprise 2 does not have any back-end application systems installed at all. Instead, all its back-end application systems are hosted at one or several ASPs. In this case the B2B protocol execution happens between enterprise 1 and an ASP that acts on behalf of enterprise 2 (Fig. 3.16).

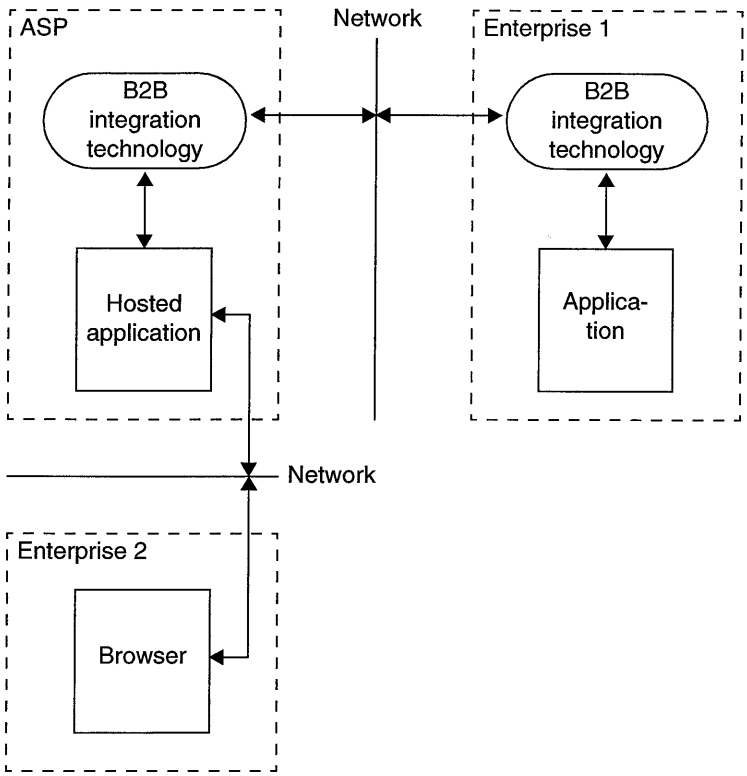

Fig. 3.16. Interenterprise integration in the case of a hosted application

Enterprise 1 integrates with enterprise 2 through a B2B protocol. Enterprise 2 has its back-end application hosted at the ASP. This results in enterprise 1 integrat-

ing with the hosted application of enterprise 2 through the ASP that hosts the application. However, enterprise 1 is not aware of this fact, since it connects to the ASP as if it were connecting to enterprise 2 (if enterprise 2 had a local deployed B2B integration technology server).

An even more interesting twist is if an enterprise that has its back-end application hosted connects to another enterprise that also has its back-end application hosted. In this case the two ASPs that host the application on behalf of the enterprises connect with each other on their behalf (Fig. 3.17).

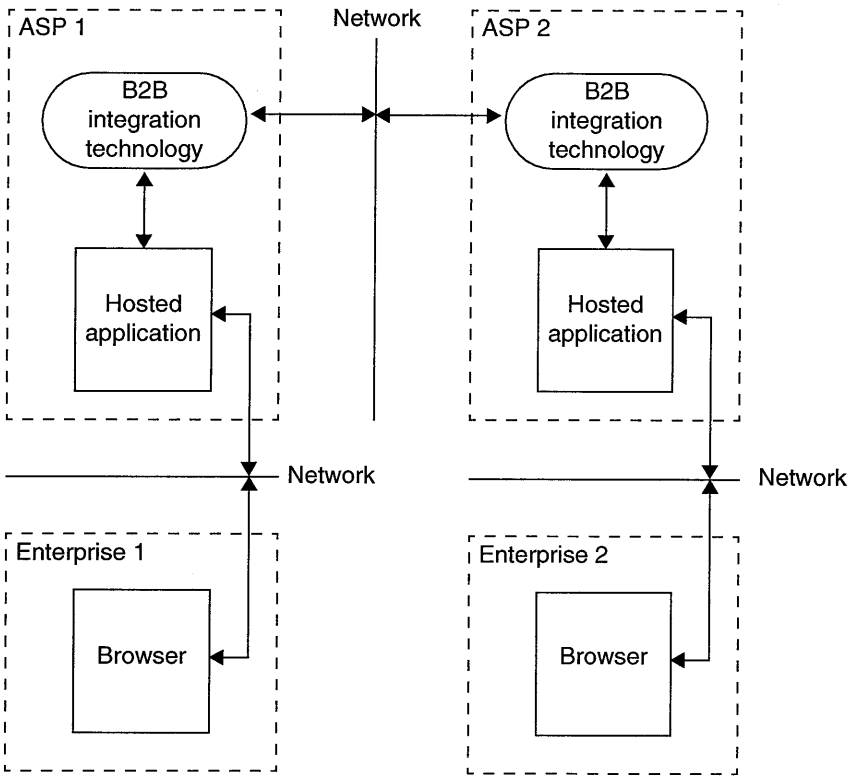

Fig. 3.17. Interenterprise integration solely through ASPs

Enterprise 1 and enterprise 2 both have their application hosted at ASP 1 and ASP 2, respectively. In order for both enterprises to interoperate and exchange business data, the ASPs hosting their applications have to communicate on the behalf of the enterprises.

Fnally, an interesting scenario arises if both enterprises have their applications hosted at the same ASP. In this case the interenterprise integration can be accomplished within the one ASP itself and no communication is required across any network. This requires that the B2B integration technology server is able to

communicate with itself. It sends out the business data to itself, and only at this point in time it does become clear that the B2B integration technology communicates with itself. Up to this point it is unaware of this, especially if there is no specific setup or "shortcut" in the B2B integration technology server itself for this case.

3.12 ASP Aggregation

Up to this point it was assumed that an ASP only hosts one back-end application system of a particular software vendor. However, this is not necessarily true. It is very well possible that an ASP hosts several different back-end application systems. All the previous discussions still apply in this case. Furthermore, within an ASP the back-end application systems can be integrated with each other using the B2B integration technology. A customer that subscribes to two back-end application systems within the same ASP can request the ASP to integrate the back-end applications with each other so that the customer's back-end application systems can communicate data between each other.

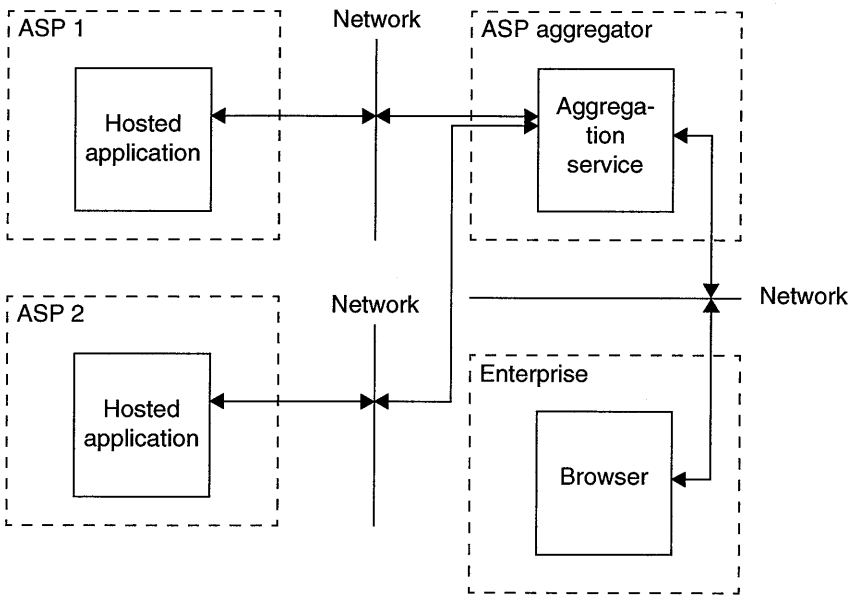

Fig. 3.18. ASP aggregation

More and more often, enterprises face the situation that the back-end application systems that they want to be hosted cannot all be hosted by the same ASP. They have to work with different ASPs. Each ASP hosts some of the back-end application systems. This requires an enterprise to coordinate with different ASPs at the same time in order to integrate their hosted applications as well as the interenterprise integration. However, it would be advantageous for these type of enterprises to have a single point of management for all their hosted applications, no matter how many ASPs host the applications. This is accomplished by ASP aggregators. These are ASPs that are gateways to other ASPs. They provide a single point of contact to enterprises that want to have their applications hosted. In turn, ASP aggregators manage the coordination between the different ASPs for an enterprise so that the enterprise gets the impression that it deals only with one ASP (Fig. 3.18).

In Fig. 3.18 the enterprise connects to the ASP aggregator. The browser of an employee has one point of contact provided by the ASP aggregator. Depending on the task the employee wants to accomplish, the ASP aggregator connects internally to the appropriate ASP that hosts the application. This architecture allows the enterprise to look at all its hosted applications in a homogeneous way as if all are installed at a single ASP.

The application integration problem also exists in the context of ASP aggregation. The various back-end application systems that an enterprise has hosted need to be integrated. In addition, the enterprise might communicate with other enterprises. Since the ASP aggregator knows about all the hosted applications of an enterprise, it can provide the integration functionality with its B2B integration technology server, and it can also provide the interenterprise integration functionality (Fig. 3.19). In this particular topology the various individual ASPs do not have to deal with integration aspects themselves. All integration functionality is provided by the ASP aggregator using its B2B integration technology server.

3.13 ASP Aggregator Aggregation

The ultimate in aggregation is ASP aggregator integration where ASP aggregators are integrated with each other to provide their subscribers with a homogeneous view across all ASPs that they are integrating. This is a necessary scenario when a subscriber's hosted applications are not covered by one ASP aggregator, but require two or more. One ASP aggregator can integrate the other aggregators to provide a homogeneous view. In addition, each ASP aggregator can integrate every other one. In this scenario all ASP aggregators can provide access to all integrated ASPs. Figure 3.20 gives an example topology where two ASP aggregators are mutually integrated with each other. The representation of ASPs in this figure is reduced and does not show the hosted applications in order to be able to fit the representation into one figure.

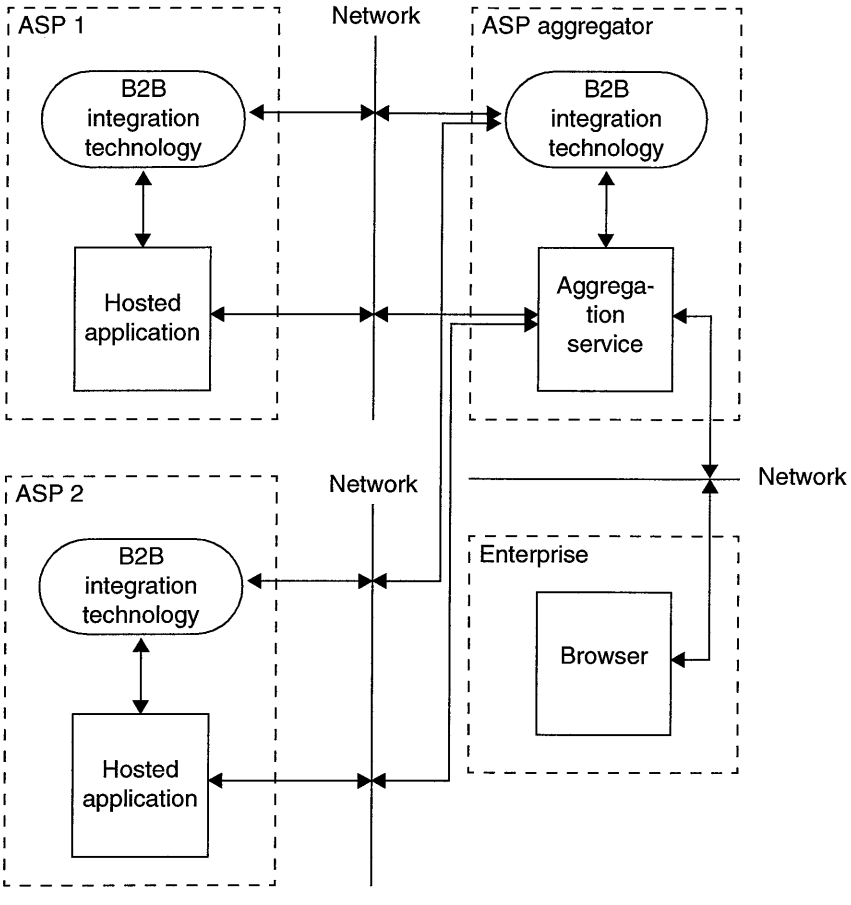

Fig. 3.19. ASP aggregator-implemented integration

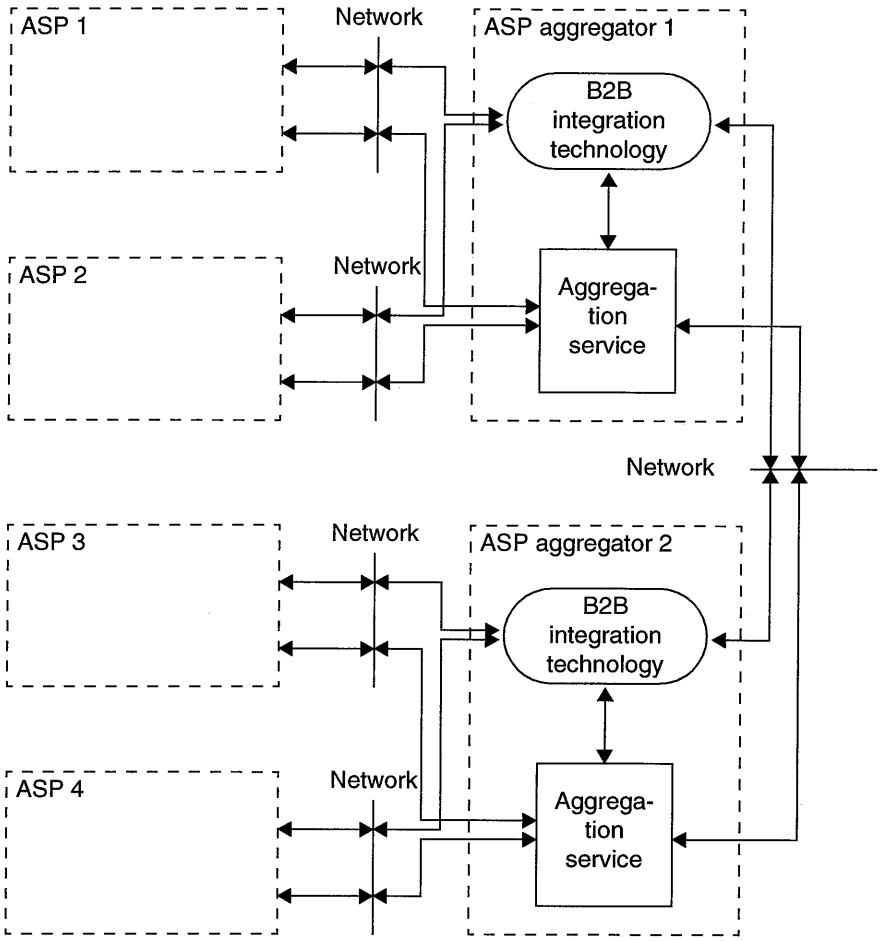

Fig. 3.20. ASP aggregator aggregation

3.14 Hosted Integration

An enterprise might not be willing to host the back-end application systems it requires. It wants its data locally stored. Instead, it would welcome an ASP hosting the integration functionality. In this case, the ASP has to worry about defining and managing the integration. Each of the back-end application systems is connected to the ASP's B2B integration technology server, in this case through dedicated B2B adapters so that the ASP can connect and integrate the back-end application systems (Fig. 3.21).

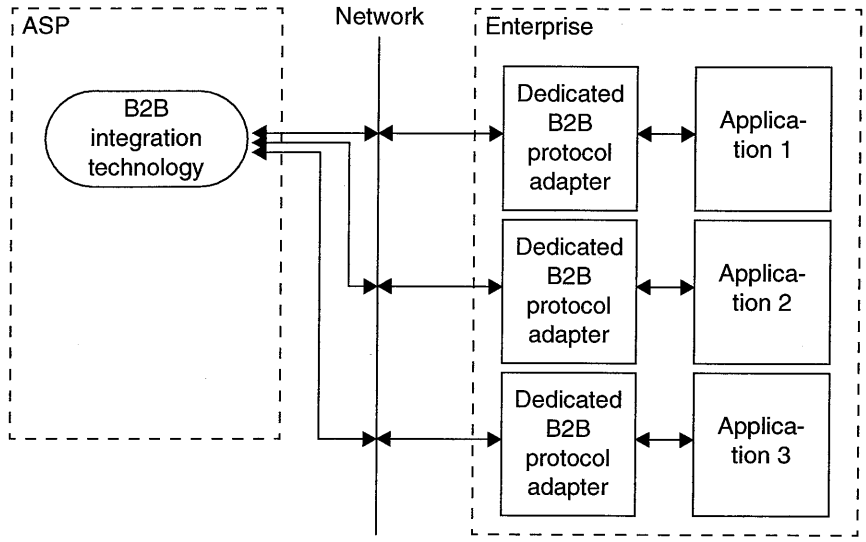

Fig. 3.21. Hosted integration

Compared to an ASP aggregator the back-end application systems are not hosted themselves, but are locally installed with the enterprise. Only the B2B integration technology server is hosted.

If a B2B integration technology server is hosted, then it must be possible to define integration for different enterprises (customers) concurrently on the same installation. If this is possible, then the intra-enterprise integration scenario discussed above can be supported in a different way. Instead of every division having to deploy and run their own B2B integration technology server, the enterprise could host one deployment internally and have the divisions act like customers that host their integration in the one deployment. This would relieve the individual divisions of the need to maintain their own B2B integration technology server installation. Furthermore, this one installation can also perform the interenterprise communication with trading partners.

3.15 Reverse Hosting

In the reverse hosting case an ASP does not install the software in-house, but at the customer's, i.e., subscriber's site. This means that the software and the data that the software manages is under full control of the subscriber; however, the management of the hosted software is performed by the ASP. This means that the subscriber out-

sources the management of the software to an ASP while still having full control over the hosted software. Figure 3.22 shows the topology of this use case. The back-end application systems integrated by the B2B integration technology at the subscriber's site are not shown.

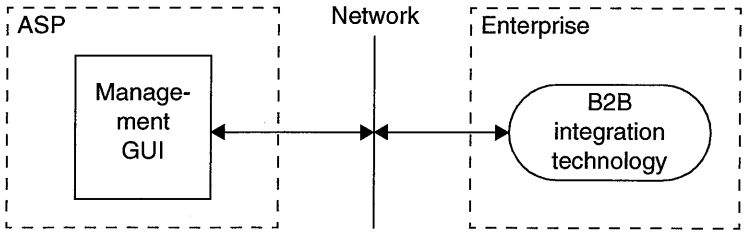

Fig. 3.22. Reverse hosting

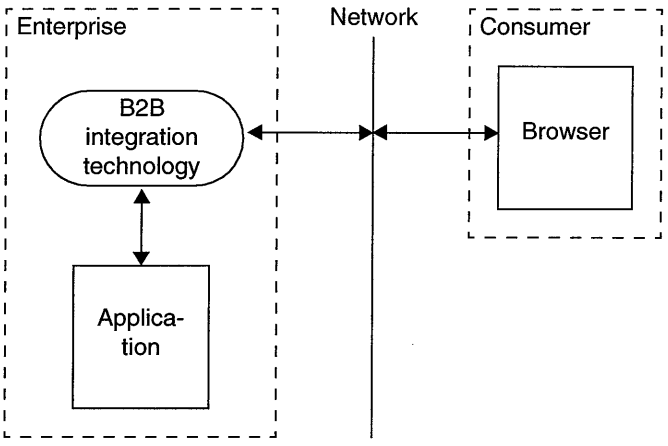

Fig. 3.23. Business-to-consumer integration

3.16 Business-to-Consumer Integration

The business-to-consumer integration scenario is very similar to the interactive application integration case discussed earlier. The difference is that consumers on their desktop or laptop computers cannot be asked to install any dedicated B2B protocol adapter to communicate. Instead, the browser the customer uses to communicate has to be directly connected with the enterprise that is selling goods. This requires the B2B integration server technology to support browser communication protocols like HTTP to be available as B2B protocol. In this case, the B2B integra-

tion technology server can directly communicate with browser technology (Fig. 3.23).

3.17 Summary

The discussed integration scenarios cover a wide range of possibilities. Some, like the hosting scenarios, seems far-fetched today, but if predictions are true that in the not-so-distant future more and more applications will be sold in hosted form, these scenarios will become increasingly common and familiar to the integration community.

From the viewpoint of this book all integration scenarios are equally important in terms of the integration concepts and the integration technology architecture. Therefore all integration scenarios will be covered by the integration concepts and the integration architecture.

http://www.springer.com/978-3-540-43487-0

B2B Integration
Concepts and Architecture
Bussler, C.
2003, XXII, 418 p., Hardcover
ISBN: 978-3-540-43487-0