MERMAID

EU Project No: IST-1999-10637

E-commerce Engine Design/Integration

Deliverable No: D7
(Rev 3)

June 2001

Report Title:	E-Commerce Engine Design/Documentation
Customer:	European Commission, Directorate-General Information Society, IST Programme
TXT Report no:	23401/D/05
Deliverable nos:	D7
Report status:	Rev 3
Date:	June 2001
Contact details:	**TXT e-Solutions S.p.A** via Frigia 27, 20126 Milano, Italy e-mail: matteo.villa@txt.it web: www.txt.it

	Name	**Signature**	**Date**
Author:	Matteo Villa		
Approved by:	Dr. Andrew Tyler		

Contributors

Company	Name
TXT	Matteo Villa
BMT	Paul Taylor
	Paul Goddard
	Chris Rawlings
CEDRE	Camille LeCat
	Michel Girin
IMGW	Bogusz Piliczewski
MO	Jack Hopkins

Distribution List

Company	Name	Number of Copies
EU Commission (DG-XIII)	Guy Weets	6
BMT	Paul Taylor	1
TXT	Matteo Villa	1
CEDRE	Vincent Gouriou	1
Met. Office	Jack Hopkins	1
IMGW-OM	Wlodek Kryzminski	1
NC	Koen DeHulsters	1

Revision

Revision Number	Date	Author	Purpose of Revision
0	23/04/2001	Matteo Villa	Initial Release
1	14/05/2001	Matteo Villa	Changes in chapters 3.1, 3.2 about system architecture
2	31/05/2001	Paul Taylor	Reformatting / corrections
3	6/06/2001	Matteo Villa, Michel Girin	Changes about pricing explanations in 1.5 and 2.1.3, 2.1.4, 2.2.1, changes in the appendix about CyberCash features

Table of Contents

1. THE MERMAID BUSINESS MODEL 6
1.1. Introduction to the Business Model 6
1.1.1. The approach 6
1.1.2. Payments 7
1.1.3. The MERMAID Data Warehouse 10
1.1.4. Risk Analysis 11
1.1.5. Summary 12
1.2. MERMAID and Data Consumers Relationship Overview 13
1.3. MERMAID and Data Providers Relationship Overview 13
1.4. Common Aspects 13
1.5. Overall Process Description 14
2. DETAILED ANALYSIS 16
2.1. MERMAID and Data Consumers 16
2.1.1. Datasets (and datastreams) pricing, including discount policies 16
2.1.2. Datasets (and datastreams) shopping basket management 18
2.1.3. Datasets (or datastreams) final price computation 18
2.1.4. On-line payment transactions 19
2.1.5. Receipts 20
2.1.6. Dataset (and datastreams) shipping 20
2.1.7. Datastreams delivery 21
2.1.8. Terms and Conditions 21
2.2. MERMAID and Data Providers 22
2.2.1. Dataset (and datastream) pricing editing including discount policies editing 22
2.2.2. Extra-costs pricing (shipping, supply medium costs) 23
2.2.3. Dataset (and datastreams) purchasing fees 24
2.2.4. Dataset purchasing data analysis 24
2.2.5. Dataset (or datastreams) warehouse space purchasing 24
2.2.6. Off-line payments 24
2.2.7. Terms and Conditions 25
2.3. Other Aspects 26
2.3.1. Currencies 26
2.3.2. Country Taxes 26
2.3.3. Startup Information 26
3. TECHNOLOGY IMPLEMENTATION 27
3.1. Introduction: the MERMAID e-Commerce Engine 27
3.2. Site Server Commerce Edition Integration 29
3.2.1. Site Server features description 29
3.2.2. Site Server architecture description 31
3.2.3. New Functionalities Required 32
3.2.4. Implementation of the New Required Features 34
3.2.5. E-Payment Integration 49
3.2.6. Integration into MERMAID architecture issues 50
3.3. Final architecture of the e-Commerce Engine 53
A. APPENDIX A: E-PAYMENT WEB SOLUTIONS 54
A.1. e-Payment Methods 54
A.1.1 Credit Cards 54
A.1.2. e-Checks 54
A.1.3. e-Cash 55
A.1.4. Smart Cards 55
A.2. e-Payment solutions over the Web analysis 55
A.2.1. CyberCash 55
A.2.2 VeriSign 56
A.2.3. CyberSource 56
A.2.4. Banca Sella 57
A.2.5. ClearCommerce 57
A.2.6. Swift 57
A.2.7. Costs Comparative Table 58

1. The MERMAID Business Model

This document describes the role of the e-Commerce facility in the MERMAID project: e-Commerce facility in the MERMAID project is supported by a complex and exhaustive business model and by a set of software tools and interfaces to support such a model. The first chapter describes an overview of the business model while the second chapter goes into a deeper detail of analysis. The third chapter describes how technology couples this model.

1.1. Introduction to the Business Model

1.1.1. The approach

The e-Commerce goal in MERMAID is to support the Data Brokering service offered by MERMAID itself. As a Data Broker, MERMAID links different data providers with different data consumers, allowing the latter to purchase the products offered by the former in a coherent, user-friendly and quick way, and therefore offering new business opportunities to data providers themselves.

The e-Commerce model developed for the MERMAID project therefore follows this precise logic that puts the Data Brokering service as the central actor of the business. In particular, the main guidelines are summarised below:

- the business development model for the project is that MERMAID will become a single business entity in its own right
- there are three kind of actors in the business model: Data Providers, selling marine and meteorological data (mainly in the form of electronic datasets but also as paper reports, etc.), Data Consumers purchasing these products and the MERMAID entity itself, selling provider's products to consumers
- all the business will happen through the MERMAID entity: no direct interaction between Data Providers and Data Consumers will be supported . Therefore the two kind of business relationships managed are "MERMAID with Data Providers" and "MERMAID with Data Consumers", as shown in figure 1 below:

Figure 1: Diagram of MERMAID Business Relationships

No direct business from Data Providers to Data Consumers will be supported, except the fact that a link to the Data Provider's site will be shown to Data Consumers, so to establish a direct link between themselves.

This does not mean that providers and consumers will not have any kind of legal agreements: MERMAID will provide two sets of Terms and Conditions, specifying the purchasing conditions and the usage conditions, so that liabilities will be established.

1.1.2. Payments

A critical point which impacts the business model is of course the way in which **payments** are made: The products MERMAID will sell do not belong to MERMAID itself. MERMAID is just selling on behalf of providers, so at the end, providers must be paid for their products.

Many possibilities exist from this point of view, each one with advantages and disadvantages. Mainly, the payment model could follow two different paths:

1- consumers pay directly to the providers
2- consumers pay to MERMAID and then MERMAID reimburses the providers

Of the two options, the one that couples more with the objective of MERMAID is the second one. In fact, not all the providers may be prepared to support e-commerce at their site (and if they were, they would be less likely to need to pass through MERMAID to sell their products anyway).

In addition, MERMAID should allow a quick and easy datasets retrieval and purchasing for customers. Therefore, it is clear that when consumers are purchasing products belonging to different providers, the easiest way is to make one only payment to MERMAID, rather than several different payments to the various providers (this would also imply a greater effort, not only in terms of time spent for the payment details and transaction, but also as additional fees to be paid for the extra transactions).

Thus the model proposed encourages consumers in the purchasing act. Now the point is not to make providers disadvantaged of this solution: providers may want to have their money back as soon as possible each time someone purchases some of their products. Also, they will want enough guarantees that MERMAID will pay them for each product sold.

Another very important point is whether to support on-line transactions or not, both towards data consumers and providers. In fact:

- **off-line transactions** are suitable where time is not a constraint, otherwise products should be sold before receiving the money, which would complicate things.
- **on-line transactions** are very straight forward but imply a much higher degree of security and therefore put more constraints on the possible payment methods accepted.

The final choice must therefore take into account consumers purchasing speed and security, providers payment and security, MERMAID business entity safety and survival (meaning that the choices made do not go against MERMAID itself - if MERMAID does not survive, all the other issues would become obsolete). In any case, MERMAID's challenge is to fully support on-line transactions, whenever possible.

In order to take a decision, a deeper analysis in terms of risks and benefits for each case is required.

1.1.2.1. MERMAID and Consumers payments:

MERMAID to Consumers payments is a typical case of a B2C (Business-to-Customer) relationship. Online transactions are only possible by relying on some web-enabled service handling the actual connection to the credit/debit institutions (banks, credit cards circuits, etc.) rather than to connect to such institutions directly.

There are plenty of sites offering such kind of services, with different payment methods supported and different running costs. All of them offer a very high degree of security both in terms of site security and connection security. Such a study is reported in the appendix of this document (examples of names are Cybercash, VerySign, Cybersource, BancaSella, etc.). In any case, the choice of which of them MERMAID actually utilises can be considered as an **exploitation** issue. The important thing is to define a set of requirements that these services must comply with, and choose the best one according to the specific needs of each MERMAID installation. Of course, the choice is not unlimited, so some kind of compromise can be foreseen.

Examples of possible requirements are:

- wide choice in the payments methods offered
- high security
- low set up and running costs
- easy integration

Therefore the payment methods supported by MERMAID will depend on the specific component chosen. It must be noticed anyway, that the only online payment method accepted world-wide and not dependent on a specific bank or country specific laws is still the **CREDIT CARD.**

Therefore, in this document, only this payment method will be taken into consideration. It is still possible however, that during the MERMAID project life-cycle or after it, certain standards may become universally accepted on the market (such as electronic Checks, e-Cash, SmartCards, etc.): this would not affect the logic of the Business Model proposed here.

Please refer to the appendix of this document for a quick overview of the various e-payment methods offered by the market, with their advantages/disadvantages.

1.1.2.2. MERMAID and Providers payments:

The case of payments between MERMAID and data providers is more open in terms of what is the best choice. Summarising the various alternatives:

- *off–line payment:* the advantage of an off-line payment case (i.e. a bank credit transfer) would be that only once every month (for example), MERMAID would have to reimburse providers all funds received from the various consumers. This situation, from the **MERMAID** point of view has the following advantages and disadvantages:

Advantages:
- the possibility of keeping money for a certain amount of time before forwarding it, with only one transaction per provider (lower transaction fees are issued)
- low security issues as providers bank details could be stored off-line

Disadvantages:
- would imply the need for an employee, or someone making the payments (higher running costs)
- possibility of providers hacking into and changing details of data sold (incorrect invoicing).
  NB. This is a security issue about accounting for the products sold (these data are needed in order to be able to give the correct amount of money back to the various providers). However, details of data sold by MERMAID will also be registered in the Internet payment sites, whose security levels are the highest, and can be consulted by MERMAID. Therefore this redundancy of data makes the issue less problematic.

From the **providers** point of view, there are also advantages and disadvantages:

Advantages:
- a safe treatment of their bank details, stored off-line
- lower administration costs because only one transaction fee per month is issued

Disadvantages:
- longer waiting time for their money
- the need to trust MERMAID to sell their data

- *on-line payment:* an on-line payment towards data providers would certainly encourage them to use MERMAID, as they would receive their money as and when their products are sold. The point is that this implies the need for more transactions, and therefore higher running costs (because of transaction fees): every time a consumer pays MERMAID for an order, this order must be split amongst the various providers whose products were present in the order. Then the money should be automatically sent to the providers. The only way in which this can be done is that MERMAID would have to give a CREDIT to providers. There are at least 2 major problems in doing this.

  Firstly, it is necessary to find a suitable way to implement on-line CREDITS, which means specific arrangements with some banks. There is not an on-line method accepted world-wide (e-checks for example, imply that both the person sending money and the person receiving money must have their bank account enabled to process e-checks). Therefore this must be presently viewed as an **exploitation issue.**

  Secondly, in order to perform such credits, MERMAID would have to store provider's accounting information, which would be a security risk for them. The only possibility would again be to rely on some specific bank service, and this should be addressed at the time of exploitation.

  From the MERMAID point of view, on-line transactions would mean lower running costs (advantage) and will not imply the need of keeping track of the various products sold data (advantage) – these data will have only an informative value, and would not be a security risk.

Of course, to solve this situation there is the possibility of using the Internet payment services described for Consumers to MERMAID payments: a new account is created for each new provider, with their bank account details (these sites usually store only 1 bank detail for each user). This would enable MERMAID to connect to the payment site either as a consumer or as a shopper and make the on-line payment. This would imply a set-up fee and monthly fees for each new provider. As a set up fee and monthly fees would approximately be $US 500 – 1000 per year, it is easily understandable that MERMAID would not be able to afford to pay for each provider. Therefore, it would be required for each provider to pay for these accounts themselves, which is likely to be a very big disadvantage for them.

In addition, as the credit card is at the moment the only universally accepted payment method for these sites, there is a high risk that MERMAID would not be able to afford to perform all the required transactions involved in reimbursing providers, especially if there is a high volume of sales and high product prices.

Again, if in the future such payment sites may offer the possibility of creating multiple accounts for the same user and more payment methods will become universally usable, this solution could become the ideal one, as it provides on-line payment without the need of storing provider's bank details

1.1.2.3. Summarising the analysed possibilities:

- off-line transaction, with the minimal risk of hacking provider's 'products sold' data
- on-line transaction with one web payment account, with the risk of hacking provider's bank account data - hardly feasible because of payment methods limitations
- on-line transaction with multiple web payment accounts - not currently feasible due to very high setup and running costs and credit limits

Therefore the proposed solution can only be the off-line transactions, with the recommendation of storing 'products sold' data both in MERMAID and Internet payment sites.

The final payment scheme therefore is:

- consumers pay the whole amount of the order to MERMAID on-line

MERMAID reimburses the providers (less the commission fee) off-line once every one month (for example)

1.1.3. The MERMAID Data Warehouse

The details described so far concern the standard process of selling environmental data products through the MERMAID site. There is another important aspect that implies payments which must be considered: the MERMAID site will offer providers the possibility to physically store their datasets in a **Data Warehouse**. This of course would imply a certain cost to MERMAID, in terms of disk space usage and administration. Therefore providers will have to pay a nominal fee in order to use such a facility.

It is important to understand that this is a separate process from the process of selling datasets, as there could be the situation in which a provider is not storing any data in the MERMAID Data Warehouse, and conversely, the situation in which a provider makes use of such a warehouse, but does not sell any datasets (because nobody is purchasing them).

Again, there are various options for the way in which these payments could be made:

- **use first, then pay:** in this logic, providers are charged a certain fee according to the physical warehouse space they actually have been using. This implies an off-line payment method, and the signing of some kind of contract in order to be able to charge money to the providers, and therefore a greater administrative effort for MERMAID. In addition, there is the risk of not being paid (and therefore going through legal processes). It would also be more difficult to manage the Data Warehouse as it would not be known in advance how much space will be required. Of course, from the provider's point of view, all this is more appealing.

- **pay first, then use:** in this case providers have to purchase the Data Warehouse space before being able to use it. This of course is more convenient to MERMAID rather than to providers, but gives the big advantage firstly of a better warehouse management (it is known in advance how much space will be required) and secondly offers the possibility of making **on-line payments** (using the same web services described in the previous chapter, so with no additional costs): providers can purchase on-line the Data Warehouse space. Of course, off-line payments are still possible, but this implies higher waiting time to providers and the need for a human operator at the MERMAID site.

It would be nice to give providers the possibility of being able to use the warehouse before paying for it, but this really implies too many risks and overheads to MERMAID, so the solution chosen is the "pay first, then use" solution, as it is a very common approach in real situations, and does not imply any real security issues.

1.1.4. Risk Analysis

To complete the overview of the business model, a very quick risk analysis is reported below. Of course the MERMAID site will use the best techniques to protect the site, however it must be remembered that selling software products on-line is always a danger. It is therefore useful to know what could happen in the case of site hacking.

The situations analysed are referred to the Business Model previously explained; therefore a model with on-line payments towards consumers (with no consumers accounting information stored by MERMAID) and off-line towards providers (with providers accounting information stored on paper), and Data Warehouse space purchased in advance.

The damage level is measured both for MERMAID and for providers, and shown in Table 1. For the situations analysed, the risk is always null for consumers.

Information Stolen or Changed	Who	Consequence	Damage		Notes
			Mermaid	Provider	
datawarehouse files (stolen)	C	consumers get datasets for free	LOW	MEDIUM	this is the least compromise providers have to pay if they want to sell their data through MERMAID
pricing / discount policy data (stolen, changed)	C, P	consumers get datasets at a cheaper price (at least zero). Providers can discover their competitors policies	LOW	MEDIUM	
commission fees values (changed)	P	providers won't pay commission fees, MERMAID will get less money	MEDIUM	NULL	
providers selling data (stolen)	P	competitors are aware of how much other providers are selling their data for	LOW	LOW	
providers selling data (changed)	P	providers can claim back more money than due. Other providers can not get their money	HIGH	MEDIUM	this would not be a problem if 'products sold' are also stored in the payment sites. In this last case, those sites are responsible for data security
datawarehouse space allocation data (changed)	P	providers get more space on the Warehouse, or space for free	LOW	LOW	

NOTE 1: C=Consumer, P=provider.
NOTE 2: Risk levels: LOW means no monetary loss but reputation loss (customers loss), i.e. someone getting something for free, MEDIUM means an amount of money that was due, but that will never arrive (i.e. a provider not being paid), and HIGH means money loss (i.e. a provider claiming back to MERMAID more money than expected)

Table 1: Site Hacking Risk Analysis

This analysis is useful when taking into account how to protect the various information in the site. Of course, the duty of the e-Commerce Business Model definition is to arrive at the best compromise in terms of user requirements, MERMAID entity convenience and minimisation of security issues, but as shown it is impossible to arrive to a situation with zero risks.

Therefore it will be a duty of site implementation to find the best way to protect the various information. In particular, a completely safe way must be found for those situations marked as HIGH risk.

1.1.5. Summary

Summarising the situation, the MERMAID business model therefore implies the establishment of two separate relationships: the one existing between MERMAID and Data Consumers, and the one existing between MERMAID and Data Providers, in addition to a set of common and administrative aspects that will be handled by MERMAID itself.

Each relationship implies a certain number of issues that must be handled, that are briefly explained in the next paragraphs. Then, in chapter 2, these relationships will be analysed in detail.

1.2. MERMAID and Data Consumers Relationship Overview

This relationship handles all the business about:

- datasets (or datastreams) pricing, including discount policies
- datasets (or datastreams) shopping basket management
- datasets (or datastreams) final price computation, including extra costs (shipping, taxes, etc.)
- on-line payment transactions
- receipts generation and storing
- dataset shipping
- shipping of an already purchased dataset (or datastream)
- Terms and Conditions

All these topics will be discussed in detail in the next chapter.

1.3. MERMAID and Data Providers Relationship Overview

This relationship handles all the business about:

- dataset (and datastream) pricing editing including discount policies editing
- extra-costs pricing (shipping, supply medium costs)
- dataset (or datastreams) warehouse space purchasing and fees
- dataset (or datastreams) purchasing fees
- providers payments
- dataset purchasing data analysis
- terms and conditions
- warehouse space purchasing and payment

All these topics will be discussed in detail in the next chapter.

1.4. Common Aspects

There are a certain number of aspects that affect both data providers and data consumers, here summarised:

- currencies
- country taxes
- site administration

These aspects will be discussed in detail in the next chapter.

1.5. Overall Process Description

We are now able to combine together all the elements described above and give an overall view of the MERMAID business process.

In general this process can be divided into the Consumer's purchasing processes and the Provider's selling process. As MERMAID is acting as a broker in between, these two processes are not synchronous: a consumer's payment to MERMAID will be reflected by one or more MERMAID payments to providers, but not at the same time. Therefore we shall discuss the two processes separately (the Data Warehouse space purchasing process can be seen as a consumer purchasing a dataset).

Purchasing Processes: the purchasing process consists mainly of data consumer's searching, selecting and paying for the datasets offered by the various providers. The site offers the possibility to search for products based on various geographic and pricing conditions. Once a consumer finds the products he needs, he can "add" this product to a virtual shopping basket. The shopping basket can contain products from different providers: all these products can then be purchased by a single operation, as the business is happening towards MERMAID (MERMAID will reimburse the various providers at a later stage). For each product it is also possible to choose between various shipping methods (including on-line methods) and supply mediums, as well as to specify different delivery addresses. The process ends with the payment confirmation; this could also start the electronic dataset delivery process, in case an electronic shipping method had been chosen.

NB. each provider can choose to sell their products, or part of them, for free. In this case no costs will be charged to the customer. Providers will be charged a minimal administration fee (see paragraph 2.2.3. about details of fees).

The second purchasing process is the Data Warehouse space purchasing process, performed by data providers (here acting as "warehouse consumers"): it is exactly as a data consumer's purchasing process, except the fact that the only "product" that is possible to buy is a disk quota for a certain amount of time. Therefore concepts such as a shopping basket and shipping costs have no sense. The only parameters are the price for disk space (per Kb for example) and the duration of such space allocation.

Selling Process: MERMAID offers to Data Provides the possibility of selling their products on behalf of them. What providers are required to do, is to fully specify their products through a metadata language, and provide details about how to physically get to these products (i.e. their web address).

Therefore the first part of the process is a set-up process: providers can add, edit or remove their products. The MERMAID site will therefore offer various options:

Geographic description: a full geographic description can be specified. This description will be according to the metadata language developed in the project (see WP3 deliverable; MERMAID Metadata Specification - D3.1) and will be used to present the product to the consumers and to allow them to search for products.

Pricing Conditions: it is possible to specify complex and flexible pricing conditions for the products, as well as discounts.

Data Warehouse: for those providers who wish to store their datasets in the MERMAID Warehouse. This implies a warehouse fee and the need of uploading the new versions.

Remote Data Access: if providers do not want to, or can not store their products in the MERMAID warehouse, it is still possible for them to automatically deliver their datasets by installing the MERMAID Remote Data Access Engine (RDAE).

Liabilities: as stated in the set of Terms and Conditions, providers will be subject to certain responsibilities.

In this document we will take into account the pricing conditions, fees and liabilities aspects as well as the payments between MERMAID and providers, but not the Data Warehouse storage or geographic description. This will be addressed in the WP3 deliverable; Data Warehouse Management (D2.1).

It must also be considered that MERMAID has to reimburse the providers for their products sold. As detailed in section 1.1.2 , this will happen in an off-line process, and therefore providers will be requested to specify their bank account details, at the time of registration, so that MERMAID will be able to reimburse them.

2. Detailed Analysis

This chapter provides a detailed explanation of all the topics the business model consists of. It is assumed that the overall e-commerce process is known, as described in the previous chapter.

All the e-commerce aspects have been grouped into the MERMAID to Data Consumers and MERMAID to Data Providers relationships, with an additional Common Aspects section, which includes aspects common to both of the two relationships.

2.1. MERMAID and Data Consumers

The relationship between MERMAID and Data Consumers gives them the possibility of purchasing one or more datasets at the same time, choosing between products from various providers, which can be purchased according to various supply and shipping options, and various payment methods. Also, the dataset price is based on a complex discount policy. In particular, the purchasing chain starts from a pricing and discount model that gives the user an individual price for each dataset. After the selection, datasets are put in the shopping basket and then, after specifying different supply and shipping options, they are paid for through an on-line transaction. This process is explained in detail below.

2.1.1. Datasets (and datastreams) pricing, including discount policies

There are different levels of pricing associated with each dataset. In particular, the pricing policy will be made with reference to a standard price to which can be applied three kinds of discounts:

- a discount based on the kind of customer
- a discount based on the kind of usage the customer is claiming to perform with the product
- a discount based on the percentage of the whole dataset that the user is purchasing (remember that MERMAID offers the possibility of purchasing subsets of large files)

In whatever scenario, the consumer will see only the actual discounted price (not the original standard price, or discounts applied), so different users may see different prices for the same dataset. Therefore the pricing policy applied by each provider will be hidden to all MERMAID users, except the provider itself.

The pricing structure from the consumer's point of view is detailed below:

- **Standard price:** is the list price of each *dataset* decided by the data provider, before applying any kind of discount or extra-cost. There is one for each dataset.

  The case of *datastreams* is slightly different. A datastream is a dataset that is regularly updated by the data provider. Typical cases are weather forecasts that may be updated on a regular basis (e.g. every 6 hours). So in the case of a datastream, we do not consider a single standard price, but rather a price for a

subscription for a certain amount of time, during which the consumer can receive all the updates. The price of this subscription will depend on the duration of the period that the consumer chooses. The pricing policies for the various periods must be decided by the data providers. So, for example, when a consumer looks for a weather forecast, they may find a certain provider offers various possibilities such as; “one day of forecasts at £10”, “7 days of forecasts at £60” or “14 days of forecasts at £110”. In this way, different subscriptions will look like different products. The advantage is that by a single purchasing operation of 14 days (for instance), the consumer will receive 14 datasets. It will then be the provider’s responsibility to send the 14 updates every day to the customer.

The standard price will be the final price only in the case when no discounts are applied. In any case, the consumer will not have the possibility of knowing if the price displayed is the standard fee, or is affected by some kind of discount; to the consumer it will be only the product price.

- **Individual discount:** for each dataset, the data provider will be able to define an individual discount, according to the specific consumer. It will be part of the data provider’s marketing policy whether to let the customer know if they are receive an individual discount or not. In any case, consumers will be aware of the fact that they can apply to a specific provider for an individual discount .

- **Category Discount:** providers may decide to apply discounts to categories of users (i.e. Universities, Research Institutes, etc.) for each dataset. The user’s category will be declared by the data consumer at the time of registration, and cannot be amended later. It is the provider’s responsibility to verify the truth of such information, as MERMAID will not be able to verify such information. In any case, it will be stated in the Terms and Conditions what kind of responsibility consumers will have in the event of providing false information.
  In the situation that a consumer has both a category discount and an individual discount, the individual discount will be applied.

- **Usage discount:** a further discount can be applied according to the kind of usage the consumer is claiming to do with the dataset he is purchasing. Consumers must claim the usage when they are searching for datasets, therefore when they find the product, the price shown will already have been discounted. The entity of this discount is decided by data providers. In addition, as MERMAID will not be able to actually verify if the consumer will be using the dataset for the purpose claimed, it will be stated in the Terms and Conditions what the responsibility the consumers will have, and it will be the providers duty to verify these conditions.

- **Subset discount:** a further discount is applied every time a data consumer decides to purchase a sub-set of a certain dataset. Data providers are able to define a discount policy for each dataset based on the percentage of the whole.

Summarising the situation consumers will have to specify a set of parameters that affect the final price:

- category: this is done only once at registration time
- usage: this must be specified at every purchase

subset: this is indirectly specified by selecting the area/time they are interested in, at every purchase. The exact percentage is then calculated by the system.

2.1.2. Datasets (and datastreams) shopping basket management

The purchasing chain has its main 'link' in the shopping basket, which is managed in the following way:

- **dataset selection:** in this phase the dataset is selected according to various methods (i.e. from a search operation, or from catalogue browsing).

- **basket adding:** once a dataset is selected, it can be added to the basket. The shopping basket is a list of all the datasets that a particular consumer is purchasing, but that he has not yet paid for. To be purchased, a dataset (or a datasetream) must be put in the shopping basket. The prices shown in the basket are already inclusive of the user's category discount and the subset discount. For datastreams it is necessary to select in advance the kind of subscription (i.e. for 3 days, for 1 month, etc.). In addition, datasets from the shopping basket can belong to whatever provider; it is not necessary that they all come from the same provider. The shopping basket is not lost if the consumer logs off from the MERMAID site. In case a provider changes the price or some discount of a product that is already present in the shopping basket before the consumer purchases the products, the consumer will be informed that such values have changed. When he finally pays for the products, the current pricing conditions will be applied, not the conditions present when he added the products to the basket.

- **basket purchasing:** a purchasing operation will purchase all the datasets present in the shopping basket. For each product the consumer will be asked to select the supply medium and shipping method from the lists available. This will allow the MERMAID site to compute the final price, including all the discounted list prices, the supply costs, the shipping costs and the taxes, as explained in the following paragraph. After that, the user is prompted with different payment methods. Only after the payment validation, products are removed from the shopping basket and put into the receipt. Also, the payment validation starts the shipping operations. It is now the provider's responsibility to couple with the correct delivery of the products.

2.1.3. Datasets (or datastreams) final price computation

The final price is calculated by adding to the discounted list price a set of other extra costs. These costs include:

- **Supply costs:** for each dataset will be included the price of the supply medium (i.e. CD, tape, etc.). The supply medium cost is defined by the data provider, who can define more than one supply medium for each dataset. In the case of multiple options, the data consumer will be able to choose the one they prefer. This price will include the handling cost for putting a dataset on a certain medium.

- **Shipping costs:** for each dataset will be included the price for the shipping (i.e. by express courier, by e-mail, etc.). The shipping cost is defined by the data provider, who can define more than one shipping method for each dataset (according to the

various supply media). In the case of multiple options, the data consumer will be able to choose the one they prefer.

- **Taxes:** a VAT tax is applied according to the country in which the dataset is shipped to. Taxes are not applied in the case of electronic shipping, as it would be impossible to determine which country the product will be shipped to. It is the consumer's liability once they receive the product, to pay the tax to their local customs.

NB: there could be cases in which the final price of the products is zero: this happens when the provider sells their products for free, and does not charge any additional costs for supply medium and shipping. In the case of a provider selling their data for free, but keeping them stored in the MERMAID warehouse, it may be that MERMAID must charge a supply and/or shipping costs, so that the final price is not zero.

2.1.4. On-line payment transactions

The last part of the purchasing operation is the on-line payment (except cases when the final price is zero). After payment verification, the product is delivered to the consumer according to the shipping methods chosen.

A consumer could theoretically choose to pay by various methods, according to the specific choice of the on-line payment provider chosen by MERMAID. As the provider may offer different methods in different countries, only the credit card, which is the only method that is universally accepted, is detailed here. For a quick analysis of the state-of-the-art payment methods, please refer to the appendix of this document.

- **credit card:** the consumer will have to provide their credit cards details (number, type, expiry date) every time they wish to finalise a purchase (note that details are not stored in the MERMAID site). These details will be passed (encrypted) to a third party entity that will perform the actual verification and debit. Various vendors are offering such payment services over the Internet, and therefore the most convenient one will be chosen (examples are: CyberCash, ICVERIFY, PayLinx, etc.).

  A transaction fee for the e-payment transaction may exist (depending on the vendors), and in that case it will be raised and charged on the final price computation. Credit card transactions usually take about 2-3 seconds to be processed.

Every time a transaction is concluded, an e-mail will be sent to the providers involved, notifying them that some of their products have been sold. They will be able to see all the details of the sale on the MERMAID site.

NB. As an example, the case of electronic checks can be considered. In this case the user should provide details about their account number and bank routing number (of course they will need an e-checks enabled bank account).

2.1.5. Receipts

A receipt is generated every time an order payment is approved. A receipt is made of:

- an unique receipt ID
- a list of all the products purchased, with the details of the shipping and supply method chosen for each of them
- the billing address
- the billing date

2.1.6. Dataset (and datastreams) shipping

Once the consumer has paid for the dataset, and MERMAID has had the opportunity to verify the validity of the transaction, a receipt is generated and the purchased products must be shipped to the consumer's various shipping addresses according to the various shipping methods selected.

This operation can be only partially automated. There are different possibilities:

1. **Dataset held by MERMAID or by a Provider who has the Data Access Engine (DAE) installed - electronic shipping method**: in this case the dataset is shipped via FTP or e-mail, which are automatically supported by the MERMAID DAE. Therefore no human intervention is required here, as all the operations are automated, including the delivery failure or success control.

2. **Dataset held by MERMAID - no electronic shipping**: in this case it is the MERMAID business entity that has the responsibility of shipping the dataset. The fact that the shipping method is not electronic implies the presence of some human operator who actually takes the dataset, puts it on some supply medium (i.e. a ZIP disk) and ships it through some courier. Of course, the costs of this operation are included in the shipping costs.

   In addition, the human operator must be aware of the fact that he has to ship something. This can be done for example, by an e-mail message sent to them fully specifying the dataset details, the supply medium and the shipping address and method.

   It is now MERMAID's liability to couple with the shipping agreement. All the possible error situations deriving from a big delay in the delivery or a missing delivery must be clearly stated in the set of Terms and Conditions about the purchase of data (see paragraph 2.1.8.).

   It must be noticed that the presence of this option is not mandatory, as MERMAID still has the possibility of supporting only electronic shipping methods, with no human intervention.

3. **Dataset held by Provider without the DAE - no electronic shipping:** this is the most complicated case. It is similar to the previous one (MERMAID shipping in a non-electronic way), but the difference is that at the provider's site someone has to be alerted about the shipping. Again, this can be done by an e-mail message specifying all the required details. It must be clearly stated in the set of Terms and

Condition who has the responsibility with regard to the shipping of the data. The structure of such an e-mail message is defined in Chapter 3.

4. **Dataset held by Provider with DAE - no electronic shipping:** this case is exactly the same as the previous one, except that the option of a dataset subset extraction is supported now. When a subset is purchased, the DAE installed at the provider's site must extract the requested subset and put it into a known location (a directory or a database). The shipping order must then also specify where the operator can find the extracted subset.

In any case, these operations will be transparent to the data consumer. The only things they need to know is the shipping address they specified (either an electronic or a physical address), which is the place where they will find the dataset, and who is responsible in the case of a failure, which will be stated in the set of Terms and Conditions.

2.1.7. Datastreams delivery

The purchasing of a datastream will usually be related to a subscription: consumers pay for a certain number of updates of a certain dataset, i.e. they pay for 20 days of weather forecasts. In this case it is clear that they should not have to pay every time they wish to receive the latest version of the product, since they have already paid at the time of subscription. Therefore, the logic is that once a consumer has paid for a subscription, it is the provider's responsibility to provide them all of the required updates that have been paid for, to the specified shipping address. All the possible conflicts and error situations deriving from this scenario (such as the case of a missing delivery) will be stated in the set of Terms and Conditions.

It should be noted that in the case of a single dataset, once the consumer has paid for it, and once they have correctly received the file, they will have to perform a new purchase operation if they require another copy of it in the future. This is because the delivery operation may imply shipping and supply costs.

2.1.8. Terms and Conditions

Every time a consumer purchases a product, he will be prompted with two sets of Terms and Conditions:

1) **Covering the actual use of the data**. This will be created by the data provider, and will cover all the aspects concerning dataset ownership, added value, royalties, etc. There will be a standard set provided by MERMAID, but the provider will be able to amend these, or even use their own, as a generic set covering all the possible situations would be impossible to produce. The standard set of Terms and Conditions will be as generic as possible, but the exact details and content will be defined at the time of exploitation, as different MERMAID services may have different conditions.

   In any case, in order to ease the purchasing process at the consumer's side, a flag showing when the sets were last updated and informing whether they are the standard set or contain variations at certain points, will be shown to the consumers when they are purchasing the products.

2) **Covering the actual purchase of the data**, which exists between MERMAID and the consumer. This will be standard and not editable, and will be produced by the MERMAID business entity. It will be shown to the consumer every time he purchases a product, and will include information such as dataset shipping and delivery liabilities, payment conditions, etc.

2.2. MERMAID and Data Providers

The relationship between MERMAID and the data provider will handle all those aspects that allow MERMAID to sell products on behalf of the providers. In general there are two cases: the first in which a data provider only sell their datasets through MERMAID and the second in which a data provider physically stores their datasets on the MERMAID site. This second case will have a certain number of implications: the provider will have to pay some warehouse fees and the choice of supply mediums and shipping methods and pricing will be decided by MERMAID itself (although the provider will be able agree or disagree with these choices).

2.2.1. Dataset (and datastream) pricing editing including discount policies editing

The pricing model at the provider's side is the same as explained in paragraph 2.1.1. The difference here is that the pricing policies can be edited for each product, so that data providers have (of course) full visibility of their policies (but not those of other providers). Also, it should be noticed that prices can be set to zero, if products are to be freely available.

In particular, noting the pricing structure, the pricing policy will be based on the standard price to which are applied the three kinds of discounts, as described in paragraph 2.1.1:

- **Standard price:** is the list price of each *dataset*, before applying any kind of discount or extra-cost. There will be one for each dataset.

  The case of *datastreams is* slightly different: in this case, we will not consider a single standard price, but the price for a subscription for a certain amount of time, during which the consumer will receive all the updates. The price of this subscription depends on the duration of the period that the consumer chooses. There can be various subscription possibilities, in order to match all the different needs. Therefore providers will be able to edit the pricing policies for the various periods.

- **Individual discount:** For each dataset, it is possible to define an "individual" discount for a specific consumer. It is the provider's decision whether to give an individual discount or not. In addition, as consumers will not be aware of the fact they are receiving any kind of discount, it is also the provider's responsibility to inform them if they are receiving such a discount. The discount is expressed as a percentage of the whole standard price.

- **Category discount:** for each dataset, the data provider will be able to define a discount based on the category of user (i.e. Universities, Research Institutes, etc.). The list of possible categories will be defined by MERMAID and will not be editable

by providers. It is the provider's responsibility to verify the authenticity of the category selected by each consumer. The consequences of a consumer falsely claiming to belong to a particular category, and purchasing products under such a claim will be stated in the set of Terms and Conditions. In the situation when a consumer qualifies for both a category and an individual discount, only the individual discount will be applied.

- **Subset discount:** a further discount is applied every time a data consumer decides to purchase a subset of a certain dataset. Providers are able to define a discount policy for each dataset based on the percentage of the whole that is actually purchased.

- **Usage Discount:** another discount will be based on the type of usage a consumer claims they will apply a specific dataset for. It is the provider's liability to verify that the consumer is actually using the dataset for the purposed claimed. The set of Terms and Conditions will state the actions to take in the case of false information.

In any situation that a discount is applied, it will be the final discounted price that is presented to the consumer, so different users may see different prices for the same dataset.

In addition, in order to ease the editing of complex pricing structures for new datasets, providers will be able to define "default" values for their products, so that every time a new dataset is created, the default values will be used to define its pricing structure. Of course, these default values can be changed at any time by the provider themselves

2.2.2. Extra-costs pricing (shipping, supply medium costs)

Every dataset (or datastream) will have different options about the supply medium on which it will be shipped and about the shipping methods. It is therefore necessary to make a distinction between datasets held by the providers and those held by the MERMAID site.

In the first case, for each dataset (or datastream) providers have the freedom of defining on which supply medium their dataset will be shipped and according to which shipping method. All the implied supply costs and shipping costs will be fully defined by the provider.

In the second case (dataset held by MERMAID) providers can not decide the different supply mediums and shipping methods - these must be decided by the MERMAID business entity. In general, MERMAID will support a "standard" set of different supply mediums and shipping methods. The costs of these will be defined by MERMAID, and will be not editable by the provider (because these are the actual shipping and supply costs that MERMAID will be supporting). Providers will only have the possibility to agree or disagree with these methods (choosing between the available options supplied by MERMAID). For example, a provider may choose not to sell a dataset in a way that may result in higher costs for their customers.

2.2.3. Dataset (and datastreams) purchasing fees

Every time a data consumer purchases a dataset from the MERMAID site, a fee is raised. The amount of this fee may vary for each provider and for each dataset and is decided by the MERMAID business entity. The fee will be based on a certain percentage of the dataset selling price (the final price, including discounts, shipping and supply costs). In any case, there will be a minimum value for this fee, so if the dataset price is so low that the percentage is less than the minimum, it will be raised the minimum value. The amount of this minimum value will be decided by the MERMAID business entity.

2.2.4. Dataset purchasing data analysis

From the MERMAID site it will be possible to have a quick view of the purchasing data. Each provider will only be able to see their data according to various ways:

1. products purchased by each consumer
2. products purchased each month / year
3. all consumers who have purchased a specific product
4. new consumers by each month / year

It will also be possible to download from the site all this as information as 'rough data' (unprocessed data - in a text format). This will allow providers to perform their own business/marketing analysis of this information.

2.2.5. Dataset (or datastreams) warehouse space purchasing

Every time a data provider decides to store their data on the MERMAID site (for whatever reason), they will need to purchase some MERMAID Warehouse space. As explained in the previous chapter, the business model will be that providers must **first pay, then use** the warehouse.

The warehouse space purchasing operation is a separate process, and is performed with on-line transactions. The price per Kb of disk space per month for the MERMAID warehouse will be the same for all providers and will be defined by MERMAID. Providers can choose how much space they require. A provider will be able to add new space if required at any time.

The space purchased will be valid for a fixed amount of time (i.e. 3/6/12 months). When the period paid for expires, the data will be removed from the warehouse by MERMAID unless an extension is purchased. Providers will be warned when their warehouse space needs renewing, and there will be a certain waiting time during which their data are still held in the warehouse.

2.2.6. Off-line payments

Every time a transaction is concluded, an e-mail will be sent to the involved providers, notifying them that one or more of their products has been sold. This data is also recorded in the MERMAID site and in the Internet payment site. Providers will have two possibilities of knowing their sales data on their site and also on the MERMAID site. MERMAID will also have the possibility of performing a double check of such information, through the

MERMAID site itself and through the Internet payment site. This information redundancy creates a higher degree of security in managing the sales data. The Terms and Conditions will state which source has the legal value in the case of a discrepancy of information from the two sources.

The payments between MERMAID and data providers are made in an “off-line” process. This means with this that data providers will not have to pay or be paid by MERMAID every time a fee is issued (when a product has been sold for example), but only at the time of the regular invoicing (i.e. once a month). This period may vary between different providers.
The site will provide statements for each provider, to show the current status of the moneys owed, when these will be paid and the next payment deadline, so that providers will know their situation.

The amount of money that MERMAID will have to pay to the various providers in a given period, will be calculated in two different ways, depending on whether the provider is using the MERMAID site for storing their datasets or not:

Datasets held by MERMAID :

For each product sold, the amount of money due to a provider will be:

= *product selling price – purchase fee – shipping and supply costs*

In this case, in fact, MERMAID is performing the shipping and providing the supply mediums, so shipping and supply costs must be not included in the money that is reimbursed to providers.

Datasets held by Provider:

For each product sold, the amount of money due to a provider will be:

= *product selling price – purchase fee*

It should be noted that in neither case a “transaction fee” is applied (the fee issued by the Internet online payment site). This is because of the following two reasons:

1- not all the payment sites raise a fee for a transaction
2- even in the case when a fee is issued, it will be only one fee per an order - which may include more than one provider’s products, and therefore can not be charged to all the providers

Therefore the “purchase fee” (or commission) charged by MERMAID, will be used to cover the transaction fee.

2.2.7. Terms and Conditions

This set of Terms and Conditions will cover *the actual use of the data*. This will be between the provider and the consumer. There will be a standard set, but the provider will be able to amend these, or use their own. Providers will be able to change their conditions at any time, and this will be covered by the Buyer Beware principal. The consumer will be shown

the Terms and Conditions at the beginning of every purchase, and all that is required is a flag showing when and where they were last updated / amended.

2.3. Other Aspects

There are a few aspects that are common to all the MERMAID business model actors, and which do not belong explicitly to either of the two relationships described before. These are the currency used for the payments and the catalogues and the cross-country taxes.

2.3.1. Currencies

MERMAID will be creating business between providers and consumers from different countries. In order to avoid problems associated with the conversion of different currencies, all the prices will be expressed in the currency of the data broker (MERMAID). This currency will be the EURO, which is the official currency of the European Union.

2.3.2. Country Taxes

In the situation that a dataset is shipped in a “physical” way (i.e. not via FTP or e-mail), a VAT tax may be applied to its price. The amount of the VAT rate for the different countries is an information that is common to all datasets and data providers, and therefore must be entered and maintained by the MERMAID business entity.

2.3.3. Startup Information

In order to start the MERMAID service smoothly from the beginning, some information will be required in the system before the launch of the service. This information will include both technical information and e-commerce information. From the e-commerce point of view, this information is:

- country taxes
- consumers categories
- dataset usage categories
- MERMAID Data Broker supported shipping methods and costs
- MERMAID Data Broker supported supply mediums and costs
- Purchasing fee values
- Warehouse space price and time validity

This information will be handled by the “site administrator” pages, which will allow the service administrator (the MERMAID business entity) to edit / insert such information.

3. Technology Implementation

This chapter will detail how the previously explained business model will be implemented into the MERMAID **e-Commerce Engine**

3.1. Introduction: the MERMAID e-Commerce Engine

The objective of the MERMAID e-Commerce Engine is to develop a software tool able to implement the MERMAID business model and at the same time to be easily integrated into the MERMAID Broker. For a full explanation of the MERMAID Broker architecture, please refer to the WP3 deliverable; MERMAID System High Level Design (D2). In general, it must be easily integrated into other architectures.

In the MERMAID architecture, the e-Commerce Engine must interface with various components (developed in WP3, WP4 and WP6), as shown in figure 2:

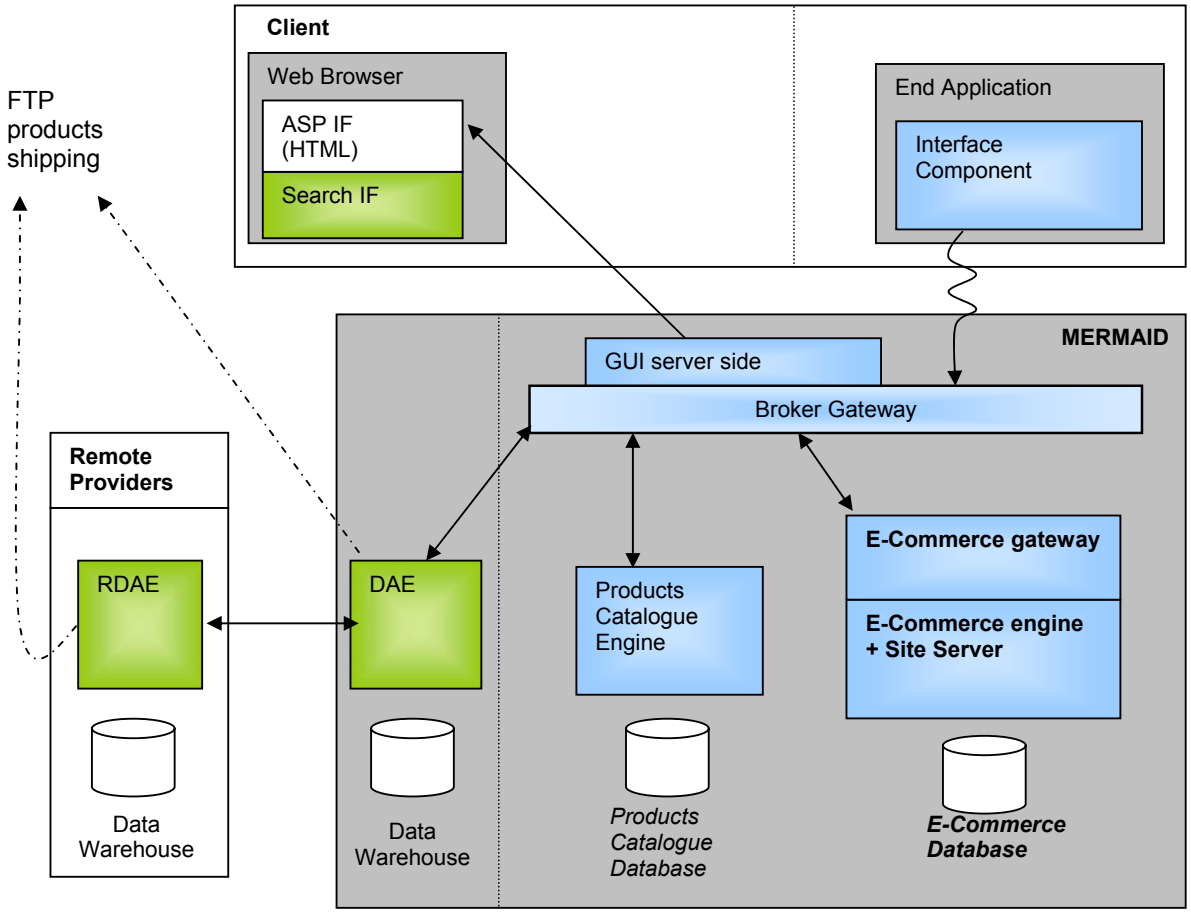


Figure 2 – MERMAID System Architecture

The role of the various components will be better explained in the WP3 deliverable – MERMAID System High Level Design (D2), but just to give a very quick overview:

Products Catalogue Engine (WP3): handles products metadata description. Allow products create, delete, update and search.

DAE / RDAE (WP3): component dedicated to access and transfer datasets from the provider site (or even MERMAID) to the end customer.

Broker Gateway (WP6): component in linking together all the other components.

Interface Component (WP4): component allowing remote applications to connect to the MERMAID site

In particular the e-Commerce Engine has the duty of driving the users in the purchasing process until the payment, then interfaces with the MERMAID Data Access Engine (DAE) to perform the dataset transfer (in case of on-line shipping).

At the provider's side, the e-Commerce Engine must cooperate with the MERMAID Products Catalogue Engine (PCE) in order to allow providers to fully describe their products (both in terms of metadata description and pricing conditions).

The complete processes are driven by the MERMAID Broker Gateway, and therefore the e-Commerce Engine must communicate with this component, which then gives it the possibility to perform all the e-Commerce functionality.

Owing to the complexity of the e-Commerce world and the great number of commercial tools available on the market, it was agreed that we should not to try to develop everything from scratch. Instead, an 'off-the-shelf' tool was chosen as a base, which could then be customised for the e-Commerce Engine development.

The final choice for this base tool was the Microsoft (MS) SiteServer Commerce Edition 3.0[1] (SSCE). This was mainly due to the very good price/features relationship: starting from a reasonable set of functionality, it has been necessary to develop other features required by the MERMAID specific needs, as explained in the next section.

The development process also took into account the development of web interfaces, database customisation and a "e-commerce gateway" layer to integrate the whole e-Commerce Engine in the MERMAID architecture.

[1] Microsoft is now selling MS Commerce Server 2000: this product is just the Commerce side of the SiteServer 3.0 Commerce edition, with few additions (i.e. more powerful wizards), therefore it's not been necessary to make a porting of the architecture

3.2. Site Server Commerce Edition Integration

The integration of MS SiteServer Commerce edition (SSCE) required the following design phases:

1- SSCE features analysis, to understand the starting point
2- SSCE architecture analysis to understand the possibilities of customisation
3- MERMAID requirements analysis, to understand which new functionality were required
4- New functionality implementation
5- e-Payment integration
6- MERMAID broker architecture integration analysis, to understand how to integrate the e-Commerce Engine
7- Final e-Commerce Engine architecture

3.2.1. Site Server features description

The following statement has been taken from a Microsoft brochure that summarises what SiteServer Commerce Edition offers:

"Microsoft Site Server Commerce Edition is a comprehensive Internet commerce server for engaging customers, transacting business, and analysing e-Commerce Web sites. Site Server Commerce Edition helps businesses deploy and manage business-to-consumer, corporate purchasing, and supply chain management applications. By providing a comprehensive set of server components, management tools, and sample sites, it significantly reduces development time and costs for these applications. Site Server Commerce Edition enables the sale of goods and services to customers and business trading partners. You can promote and merchandise products dynamically, run a more efficient online business, derive revenue from online advertising, and understand and improve your business through comprehensive site analysis.

Site Server Commerce Edition provides a comprehensive set of features that can be easily integrated into existing accounting or order management systems. And software from more than 50 independent application vendors makes it possible to extend the platform—with specialized billing, payment, or accounting systems, for example—with less need for custom development"

Table 2 highlights the various functionality offered.

Commerce Server	Commerce Server includes a comprehensive set of features, sample sites, and tools that enable you to engage customers and transact business online: – **Site Builder Wizard and Sample Sites** remove the complexity of database schema editing, scripting, and HTML coding. The simple, step-by-step approach of the Site Builder Wizard dramatically reduces site development time. Sample sites provide examples of business-to-consumer and business-to-business commerce applications. – **Dynamic merchandising** provides support for easy, real-time administration of product and price promotions from any remote Web browser through the Promotion Wizard. Intelligent CrossSell uses previous shopper trends to automatically make recommendations. – **Order Processing Pipeline** handles targeted functions—such as product tax, shipping and handling charges, payment authorizations, and inventory checks—according to specific business rules. It can be integrated with existing systems and can also be extended with many products from independent software vendors. – **Commerce Interchange Pipeline** enables applications to exchange information using the Internet or an existing EDI system. Because the Commerce Interchange Pipeline is data format-independent and transport-independent, businesses of all sizes can communicate securely. The Commerce Interchange Pipeline supports native Web formats such as XML and HTTP as well as those from numerous independent EDI software vendors. – **Integration with Microsoft Transaction Server—**a transaction processing system included with the Windows NT Option Pack—allows Site Server to provide site and application developers a business-critical solution that offers significantly higher reliability in business transactions. – **Dynamic catalogue generation** creates custom Web catalogue pages on the fly using Active Server Pages to directly address the needs, qualifications, and interests of visiting customers. – **Rich object model** manages products, users, and orders. It supports schema and database independence, enabling businesses to integrate their existing business rules and data with their online presence. – **Buy Now,** a powerful online marketing solution, lets you embed product information and order forms in most online contexts—such as online banner ads—for quick, spontaneous purchase by consumers. – **Commerce Host Administrator,** a control center for site administrators and Internet hosting service providers, enables the centralized administration of multiple commerce sites while allowing individual site managers to update their sites remotely – **Built-in Microsoft Wallet** support helps businesses provide customers with the most convenient and secure online payment experience. – **Industry-standard security** creates a secure environment for customers, partners, and site/application administrators with strong, integrated HTTP Authentication and Windows Challenge Response. Site Server Commerce Edition supports real-time credit authorization with secure transaction protocols such as SSL and Secure Electronic Transaction (SET). **Commerce Server Software Developer's Kit (SDK),** a set of open application programming interfaces (APIs), allows full extensibility across the entire Order Processing and Commerce Interchange Pipelines.

Table 2: Functionality offered by MS Site Server Commerce Edition 3

Personalisation and Membership	Membership lets you easily manage users and user profiles for high-volume sites. Secure access to any area of the site, supporting subscription or "members only" applications. Personalisation enables the delivery of custom content based on the site visitor's personal profile and supports targeted promotions and one-to-one marketing. – **Direct Mailer** is an easy-to-use tool for creating a personalized direct e-mail marketing campaign based on Web visitor profiles and preferences. – **Membership Server** provides the software infrastructure for efficiently managing secure access to Web sites and site content, with the ability to scale to support millions of visitors. Authentication can be based on cookies, Basic and HTML forms, challenge/response, and certificates.
Ad Server	Ad Server manages ad schedules, customers, and campaigns through a centralized, Web-based management tool. Target advertising to site visitors based on interest, time of day or week, and content. In addition to providing a potential source of revenue, ads can be integrated directly into Commerce Server for direct selling or lead generation.
Site Server Analysis	The Site Server Analysis tools let you create custom reports for in-depth analysis of site usage data. Create industry-standard advertising reports to meet site advertiser requirements. Classify and integrate other information with Web site usage data to get a more complete and meaningful profile of your visitors and their behavior. Enterprise management capabilities enable the central administration of complex, multi-homed, or distributed server environments. Supports 28 Web server log file formats on Windows NT, UNIX, and Macintosh operating systems, including those from Microsoft, Netscape, Apache, and O'Reilly. – **Commerce Order Manager** gives direct access to real-time sales data on your site. Analyse sales by product or by customer to provide insight into current sales trends or manage customer service. Allow customers to view their order history online.

Table 2 (cont.): Functionality offered by MS Site Server Commerce Edition 3

3.2.2. Site Server architecture description

As detailed in the previous section, SSCE is a set of tools to be used to build a commerce site rather than a specific tool in itself. Of course, site wizards provide a quick start through the creation of standard sites. The following section provides an analysis of these objects and their architecture, using the diagram in Figure 3 as a reference.

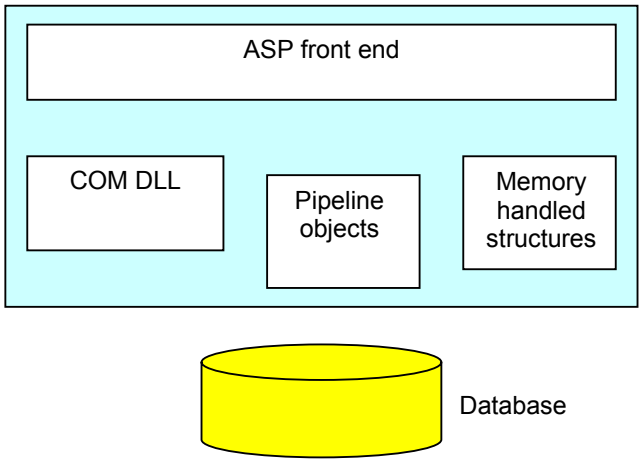


Figure 3: Diagram showing architecture of the SSCE Objects

The core is made by a set of **COM DLL objects**. These libraries provide several functions such as: page layout, database storage mapping, data functions, order form handling, file documents handling, dictionaries, pipelines controls, etc.

These COM objects can be used by an **ASP front end**: ASP (Advanced Server Pages) is a Microsoft format for web pages, which is made by a mix of HTML with VB scripts code. Therefore COM objects can be used from within a web page.

Through these COM objects it is also possible to access the underlying Database (via ADODB). This database contains all the information about shoppers, products catalogue, pricing conditions, discounts, etc.

Finally, **Pipelines** are the central concept of SiteServer: a "pipeline" is a series of COM objects executed in cascade, manipulating the same memory structures and performing various actions.

Typical examples of pipelines are:

*Order Pipeli*ne: performs the validation and fulfillment of an order, by checking required fields and calculating others (i.e. discounts, shipping costs…)

Purchase Pipeline: performs the purchasing phase of an order, by interfacing to a web payment system and generating receipts.

The advantage of SSCE pipelines is that each pipeline can be made by various pipeline components and configured at run time. Also, each component can be customised, developed from scratch or even purchased by third parties (typical examples are components to connect to the web payment servers), but a pipeline is a single logical process. SSCE provides editors to graphically edit and control pipelines.

The most important object that is passed through the various stages of a pipeline is the "**Order Form**". This is an object held into memory (then saved to a binary field in the database) containing all the information about an order; such as customer billing address; shipping address; various items (an order can be made by one or more items); item prices; with and without discounts, etc.

3.2.3. New Functionality Required

Even if SSCE offers a set of tools rather than an embedded solution, analysing the features offered by SSCE and the MERMAID business model, it is easily understandable that many concepts were not taken into account and therefore are missing to SSCE. In particular:

Business model: the SSCE typical case is a shop selling its products on-line (B2C). The MERMAID scenario is that of a "data broker", selling products on behalf of other business entities (data providers). The first need is therefore to distinguish between Consumers (typical SSCE users) and Providers. Then products must be associated to those providers.

Metadata: the second important need not covered by SSCE is the complex metadata description: oceanographic/meteorological datasets require complex descriptions (as

defined in the WP3 deliverable; MERMAID Metadata Specification - D3.1) not considered by SSCE.

Discounts: there is nothing in SSCE about discounts that are calculated at purchasing time, as subset discounts and usage discounts (SSCE discounts are based on “fixed” values and based on particular products or consumers).

Users and logins: it must be noticed that in SSCE there were only two types of users: shoppers (consumers) and Site Administrators (i.e. shop owner), able to modify all products pricing conditions and descriptions.

In our case, the product can only be modified by the provider who owns them. Therefore in MERMAID there are three different type of users: consumers, providers and MERMAID administrators (only able to modify site common aspects) and of course, all these users need different logins, different homepages and profiling.

Orders structure: a big modification is then required in the way orders are processed: in SSCE orders can have multiple items, but all items belong to the same shop and they are all shipped to the same address. In MERMAID, every order can contain different items belonging to different providers, and these items can be shipped to different locations.

Datastreams: the concept of a subscription to receive updates of a certain product for a certain amount of time is totally missing in SSCE.

Extra costs: SSCE only can associate shipping costs and handling costs to one order, while for MERMAID there could be different shipping costs for each item and supply costs. In particular these costs are decided by each provider and can be different for each one of their products. Also, for each product there could be different supply mediums and shipping methods.

Manual shipping: another need is the one to advise providers in case of manual shipping: an e-mail must be sent to them.

Providers payment: all the business towards providers is not considered in SSCE: the amount of money due to the different providers and the warehouse space purchasing are new requirements.

New tables: new tables are also needed to store the information not directly expressed in the previous points but needed in order to correctly support all this, such as users categories, country taxes, user roles, etc.

Integration: from the software point of view an “open architecture” is required in order to be integrated with the other components of the MERMAID broker. SSCE architecture just provides a set of tools that can be used in various ways. Therefore a “common gateway” is needed in order to provide a precise and robust interface to the e-Commerce Engine, especially for those processes that involve other MERMAID components.

3.2.4. Implementation of the New Required Features

In general there are five elements in SSCE that can be created or modified in order to fulfill the new MERMAID required features. These are:

- DataBase changes: tables / fields / relationships
- COM DLL: providing new complex functionality
- Memory structures: objects such as the order form are created and handled into memory, then stored into binary fields of the database
- Pipelines customisation and new component-adding
- ASP pages: these are the Web front end, and should not contain too much logic (business logic should be inside of COM objects). In any case new interfaces are required for new functionality

It should be noted here that during the MERMAID Broker development, it may be necessary to change, create, or delete some of the e-Commerce functionality. In order to match integration needs, user requirements, bugs fixing or whatever need that could emerge during a standard software development process.

For this reason only the most important software developments are detailed below.

3.2.4.1. COM OBJECTS

An example of the typical EcommerceGateway classes are provided:

ECommerceGateway.Discount: this class provides a method to calculate the discounted price of an object:

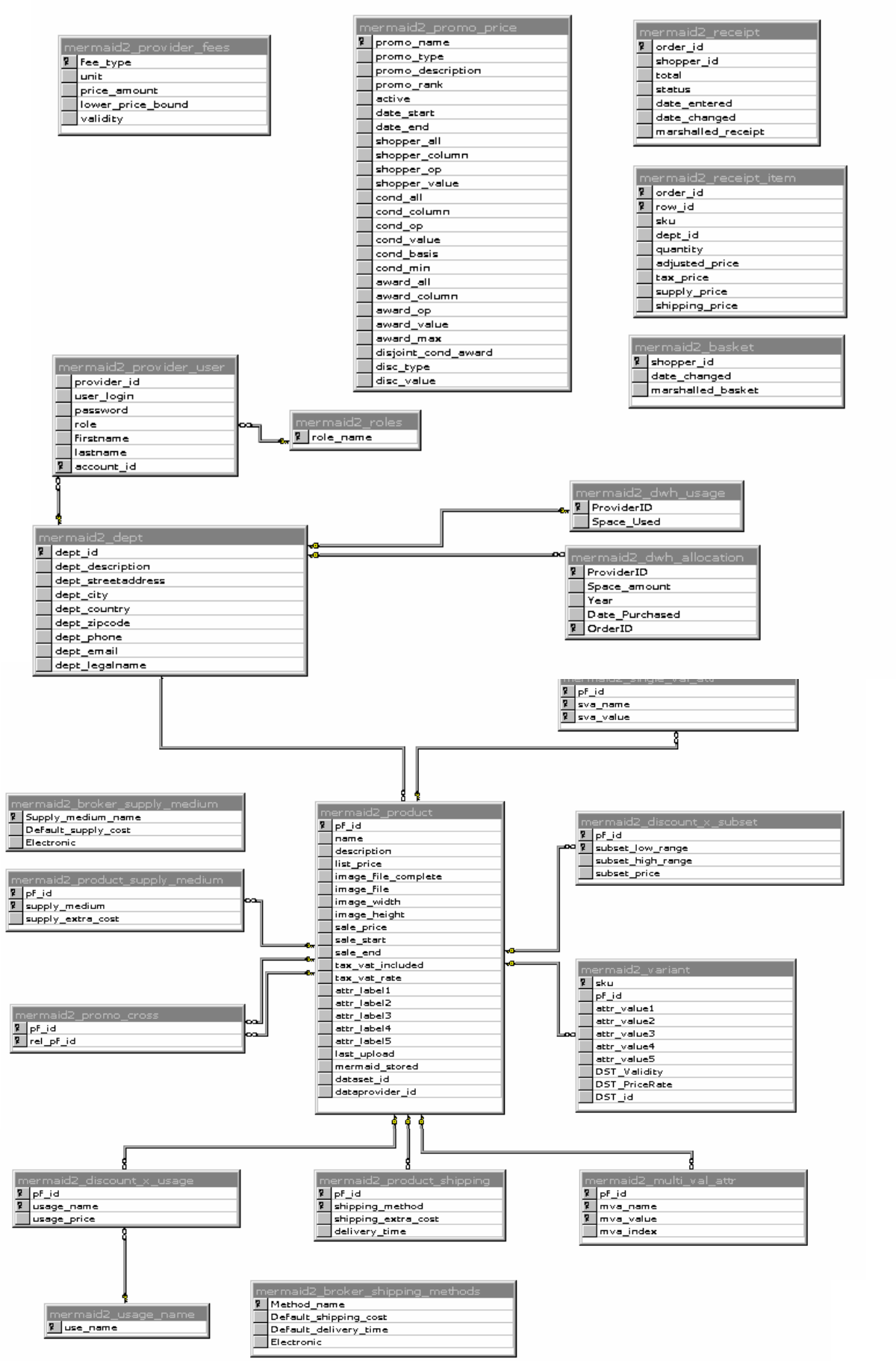
GetDiscountedPrice (prod_id, cmdTemp, shopper_id, use, subset, errorList)
where
prod_id = MERMAID product ID
cmdTemp = DB command
shopper_id = MERMAID consumer's ID
use = type of usage the user wishes to perform
subset = percentage of the whole subset the user is going to purchase
errorList = list object to handle errors
(all parameters of type "variant")

ECommerceGateway.Sender: this class will send a message after a purchase operation has been finalised to notify the provider that a certain dataset must be shipped to a specific address. The receiver of the message can be either a human provider (via e-mail) or the MERMAID Data Access Engine (via the MERMAID communication infrastructure).

ECommerceGateway.DWH: this class provides functions to manage the space that each provider purchases on the MERMAID DataWarehouse (get, allocate, de-allocate, etc.)

3.2.4.2. DB SCHEMA

The e-Commerce DataBase (DB) schema is represented in Figure 4. A full description of the tables and meanings of the fields shown in the schema is also provided (fields in *italic* are primary key, fields sizes are reported only for 'char' and 'varchar' types).


mermaid2_provider_fees
fee_type
unit
price_amount
lower_price_bound
validity
mermaid2_promo_price
promo_name
promo_type
promo_description
promo_rank
active
date_start
date_end
shopper_all
shopper_column
shopper_op
shopper_value
cond_all
cond_column
cond_op
cond_value
cond_basis
cond_min
award_all
award_column
award_op
award_value
award_max
disjoint_cond_award
disc_type
disc_value
mermaid2_receipt
order_id
shopper_id
total
status
date_entered
date_changed
marshalled_receipt
mermaid2_receipt_item
order_id
row_id
sku
dept_id
quantity
adjusted_price
tax_price
supply_price
shipping_price
mermaid2_basket
shopper_id
date_changed
marshalled_basket
mermaid2_provider_user
provider_id
user_login
password
role
firstname
lastname
account_id
mermaid2_roles
role_name
mermaid2_dwh_usage
ProviderID
Space_Used
mermaid2_dept
dept_id
dept_description
dept_streetaddress
dept_city
dept_country
dept_zipcode
dept_phone
dept_email
dept_legalname
mermaid2_dwh_allocation
ProviderID
Space_amount
Year
Date_Purchased
OrderID
pf_id
sva_name
sva_value
mermaid2_broker_supply_medium
Supply_medium_name
Default_supply_cost
Electronic
mermaid2_product_supply_medium
pf_id
supply_medium
supply_extra_cost
mermaid2_promo_cross
pf_id
rel_pf_id
mermaid2_product
pf_id
name
description
list_price
image_file_complete
image_file
image_width
image_height
sale_price
sale_start
sale_end
tax_vat_included
tax_vat_rate
attr_label1
attr_label2
attr_label3
attr_label4
attr_label5
last_upload
mermaid_stored
dataset_id
dataprovider_id
mermaid2_discount_x_subset
pf_id
subset_low_range
subset_high_range
subset_price
mermaid2_variant
sku
pf_id
attr_value1
attr_value2
attr_value3
attr_value4
attr_value5
DST_Validity
DST_PriceRate
DST_id
mermaid2_discount_x_usage
pf_id
usage_name
usage_price
mermaid2_product_shipping
pf_id
shipping_method
shipping_extra_cost
delivery_time
mermaid2_multi_val_attr
pf_id
mva_name
mva_value
mva_index
mermaid2_usage_name
use_name
mermaid2_broker_shipping_methods
Method_name
Default_shipping_cost
Default_delivery_time
Electronic

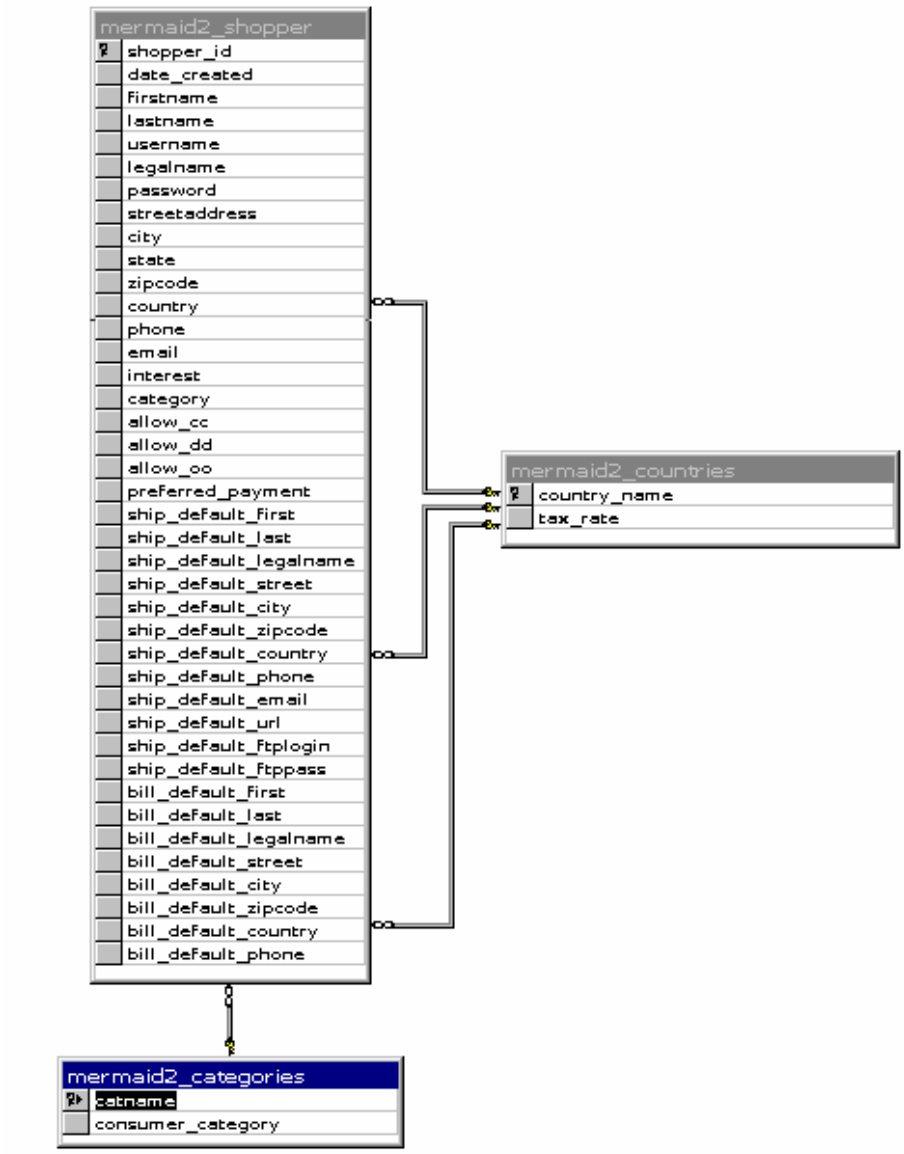


Figure 4: MERMAID e-Commerce Database Schema

The most important tables are detailed here:

_Basket
This table stores the consumer's basket. It's content is updated every time a user selects a new dataset from the catalogue, and is deleted when the user purchases the products.

Field Name	Data Type	Required	Notes
shopper_id	char (32)	Y	user's ID
date_changed	datetime	Y	last change
marshalled_basket	image	Y	binary column containing the whole shopping form (described in this paragraph)

_Broker_Shipping_Methods
This table contains the list of the shipping methods accepted by the MERMAID Broker, with their prices. These values are presented to the provider so that they can select which of these shipping methods to associate with each product (they cannot edit the values however).

Field Name	Data Type	Required	Notes
Method_name	varchar (32)	Y	name of the shipping method
Default_shipping_cost	int	N	default cost
Default_delivery_time	varchar (20)	N	default delivery time
Electronic	char (1)	Y	'Y' if this is an electronic shipping method

_Broker_Supply_Medium
This table is similar to _BROKER_SHIPPING_METHODS but relates to the dataset's supply media.

Field Name	Data Type	Required	Notes
Supply_medium_name	varchar (32)	Y	name of the supply medium
Default_supply_cost	int	N	default cost for that medium
Electronic	char (1)	Y	'Y' if it is an electronic medium (for electronic shippings)

_Categories
This table stores the list of the consumers categories.

Field Name	Data Type	Required	Notes
catname	varchar (100)	Y	name of the category
consumer_category	char (1)	N	'n' if not a consumer category

_Countries
This table manages country-specific information. Currently, the only data management is the value of the VAT taxes, to be used when calculating the shipping costs.

Field Name	Data Type	Required	Notes
country_name	varchar (100)	Y	Name of the country
tax_rate	int	N	VAT value (percentage)

_Dept
This table is the main table to store Data Provider's profiles (the name comes from the MS SSCE concept of "department"). Note that there can be different users for each Provider entity, with different roles in the organisation, and therefore with different access policies: these are stored in table "_provider_user"

Field Name	Data Type	Required	Notes
dept_id	char (32)	Y	Dataprovider unique ID
dept_description	varchar (255)	N	Description field
dept_streetaddress	varchar (100)	N	Legal address of the provider
dept_city	varchar (100)	N	…
dept_country	varchar (100)	N	…
dept_zipcode	varchar (100)	N	…
dept_phone	varchar (100)	N	…
dept_email	varchar (100)	N	personal (or corporate) e-mail
dept_legalname	varchar (50)	N	Provider legal name

_Discount_x_Subset
This table stores the amount of discount to be applied to a specific product according to the percentage of the whole file that the user is purchasing.

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	Y	product ID
subset_low_range	int	Y	subset range low value
subset_high_range	int	Y	subset range high value
subset_price	int	Y	discounted price for the given range (% of the whole)

_Discount_x_Usage
This table stores the values of the discounts to be applied depending on the type of usage the consumer is going to perform with the dataset.

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	Y	MERMAID product ID
usage_name	varchar (100)	Y	Name of the usage type
usage_price	int	Y	Discounted price (% of the whole)

_Dwh_allocation
This table is used to store the amount of DataWarehouse space that providers have purchased

Field Name	Data Type	Required	Notes
ProviderID	char (32)	Y	Provider unique ID
OrderID	varchar (32)	Y	Purchasing order ID
Space_Amount	int	Y	amount of space (in MB)
Year	int	Y	year in which the space is valid
Date_Purchased	datetime	N	purchasing date

_Dwh_usage
This table handles the MERMAID Warehouse space that's been actually used by a provider

Field Name	Data Type	Required	Notes
ProviderID	char (32)	Y	Provider unique ID
Space_Used	int	Y	Amount of space used (in MB)

_Multi_Val_Attr
This table is automatically created by Site Server and contains the product multi-value attributes.

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	yes	Unique product identifier
mva_name	varchar (100)	yes	Attribute's name
mva_value	varchar (100)	yes	Attribute's value
mva_index	int	no	Attribute index

_Product
This table contains all the products information that are relevant for the e-Commerce Engine (NB. product metadata description is stored in the MERMAID Catalogue Database).

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	yes	Unique product identifier
name	varchar (255)	no	Product information
description	varchar (255)	no	…
list_price	int	no	basic price
image_file_complete	int	no	preview image information
image_file	varchar (255)	no	…
image_width	int	no	…
image_height	int	no	…
sale_price	int	no	product sale price and …
sale_start	datetime	no	start
sale_end	datetime	no	end dates
tax_vat_included	tinyint	no	…
tax_vat_rate	real	no	…
attr_label1	varchar (100)	no	names of the multi-value attributes
attr_label2	varchar (100)	no	…
attr_label3	varchar (100)	no	…
attr_label4	varchar (100)	no	…
attr_label5	varchar (100)	no	…
last_upload	datetime	no	data of product last upload
Mermaid_stored	char (1)	no	'Y' if the product is stored in the MERMAID DataWarehouse
dataset_id	varchar (100)	no	Product ID as used by owning Provider
dataprovider_id	char (32)	no	Provider's ID

_Product_Shipping
This table contains all the shipping methods, costs and delivery times that the consumer can choose for a given product.

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	yes	Unique product identifier
shipping_method	varchar (32)	yes	shipping method name
shipping_extra_cost	int	no	Shipping method price
delivery_time	varchar (20)	no	Expected delivery time

_Product_Supply_Medium
This table contains the supply media and costs that the consumer can choose for a given product.

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	yes	Unique product identifier
supply_medium	varchar (32)	yes	Supply kind
supply_extra_cost	Int	no	Supply price

_Promo_Cross
This table is automatically created by Site Server and contains information about special promotions (i.e. if consumer purchases a product that is in MERMAID2_promo_cross it will has the opportunity to purchase another product, on basis of promotion, at a suitable price. This will currently not be used in MERMAID.

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	yes	Unique product identifier
rel_pf_id	varchar (100)	yes	Linked price for promotion

_Promo_Price

This table is automatically created by Site Server and is used to compute various product's discounts. In MERMAID it is only used for consumer and/or category discounts.

Field Name	Data Type	Required	Notes
promo_name	varchar (100)	yes	Unique promotion identifier
promo_type	int	yes	Fixed value "100"
promo_description	varchar (255)	no	NULL
promo_rank	int	no	"10" for category, "50" for individual discounts
active	int	no	"1"
date_start	datetime	no	start and …
date_end	datetime	no	end dates
shopper_all	int	no	"0"
shopper_column	varchar (64)	no	"shopper_id" or "category"
shopper_op	varchar (2)	no	"="
shopper_value	varchar (64)	no	shopper's category or id
cond_all	int	no	"0"
cond_column	varchar (64)	no	"_product_pf_id"
cond_op	varchar (2)	no	"="
cond_value	varchar(64)	no	pf_id (product id)
cond_basis	char (1)	no	"Q"
cond_min	int	no	"1"
award_all	int	no	"0"
award_column	varchar (64)	no	"_product_pf_id"
award_op	varchar (2)	no	"="
award_value	varchar (64)	no	pf_id (product id)
award_max	int	no	"1"
disjoint_cond_award	int	no	"0"
disc_type	char (1)	no	"%"
disc_value	real	no	discount amount (%)

_Provider_User

This table contains the profiles of the various users belonging to the same provider: different profiles are required in order to handle different access policies.

Field Name	Data Type	Required	Notes
account_id	char (32)	yes	Unique identifier of the provider user
provider_id	char (32)	yes	foreign key: provider ID
user_login	varchar (255)	yes	user login…
password	varchar (50)	yes	.. and password
role	varchar (32)	yes	User role (see table _roles)
firstname	varchar (50)	yes	First name
lastname	varchar (50)	yes	…and last name

_Provider_Fees

This table contains details of MERMAID fees on the basis of purchase or space disk that the provider would allocate.

Field Name	Data Type	Required	Notes
fee_type	varchar (50)	yes	Fee type (datawarehouse or purchase)
unit	varchar (20)	no	Monetary fee or space disk allocate
amount	money	yes	Lower amount
lower_price_bound	money	no	Lower MERMAID fee

_Receipt
This table stores consumer's orders and is filled when he purchases an order form (data are passed from table "_basket" to table "_receipt").

Field Name	Data Type	Required	Notes
order_id	char (32)	yes	Unique order identifier
shopper_id	char (32)	yes	Unique consumer identifier
total	int	no	Receipt total
status	int	no	Receipt status
date_entered	datetime	no	Date entered
date_changed	datetime	no	Data changed
marshalled_receipt	image	no	Binary field containing relevant information from the purchased order form

_Receipt_Item
This table contains items level information for each order in the receipt. (see table "_receipt").

Field Name	Data Type	Required	Notes
order_id	char (32)	yes	Unique order identifier
row_id	int	yes	Row id
sku	varchar (100)	no	Unique product sub-identifier
dept_id	char (32)	no	Unique provider identifier
quantity	int	yes	Quantity of product (not used)
adjusted_price	int	yes	Price information
tax_price	int	no	…
supply_price	int	no	…
shipping_price	int	no	…

_Role
This table contains the role list that MERMAID assigns at consumer.

Field Name	Data Type	Required	Notes
role_name	varchar (32)	y	Role name

_Shopper
This is the main storage for Data Consumer's profiles.

Field Name	Data Type	Required	Notes
shopper_id	char (32)	Y	Shopper identifier create by MERMAID
date_created	datetime	Y	Date of shopper login creation
firstname	varchar (100)	N	Consumer name
lastname	varchar (100)	N	.. and last name
username	varchar (100)	N	MERMAID login name
legalname	varchar (100)	N	Consumer's legal name
password	varchar (100)	N	MERMAID password
streetaddress	varchar (100)	N	Consumer Legal address
city	varchar (100)	N	…
state	varchar (100)	N	…
zipcode	varchar (100)	N	…
country	varchar (100)	N	…
phone	varchar (100)	N	…
email	varchar (100)	Y	consumer's email address
interest	varchar (100)	N	generic interests
category	varchar (100)	N	consumer's category (see table _categories)
allow_cc	char (1)	N	Allow credit card
allow_dd	char (1)	N	Allow other payment method
preferred_payment	char (2)	N	Consumer's preferred payment method
ship_default_first	varchar (100)	N	Default ship information
ship_default_last	varchar (100)	N	…
ship_default_legalname	varchar (100)	N	…
ship_default_street	varchar (100)	N	…
ship_default_city	varchar (100)	N	…
ship_default_zipcode	varchar (100)	N	…
ship_default_country	varchar (100)	N	…
ship_default_phone	varchar (100)	N	…
ship_default_email	varchar (100)	N	…
ship_default_url	varchar (100)	N	…
ship_default_ftplogin	varchar (100)	N	…
ship_default_ftppass	varchar (100)	N	…
bill_default_first	varchar (100)	N	Default bill to information
bill_default_last	varchar (100)	N	…
bill_default_legalname	varchar (100)	N	…
bill_default_street	varchar (100)	N	…
bill_default_city	varchar (100)	N	…
bill_default_zipcode	varchar (100)	N	…
bill_default_country	varchar (100)	N	…
bill_default_phone	varchar (100)	N	…

_Single_Val_Attr
This table is automatically created by MERMAID and contains the product single attributes.

Field Name	Data Type	Required	Notes
pf_id	varchar (100)	no	Identifier number created by MERMAID
sva_name	varchar (100)	no	Single attribute name
sva_value	varchar (100)	no	Single attribute value

_Usage_Name
This table contains the possible dataset usage names.

Field Name	Data Type	Required	Notes
use_name	varchar (100)	Y	usage way list

_Variant
This table is automatically created by Site Server, and stores product variant attribute values (variant attributes names are in table "_product"). It is also used to manage datastreams pricing.

Field Name	Data Type	Required	Notes
sku	varchar (100)	Y	Product sub-identifier number
pf_id	varchar (100)	Y	MERMAID product ID
attr_value1	varchar (100)	N	Product attribute value
attr_value2	varchar (100)	N	Product attribute
attr_value3	varchar (100)	N	Product attribute
attr_value4	varchar (100)	N	Product attribute
attr_value5	varchar (100)	N	Product attribute
DST_Validity	int	N	Datastream validity
DST_PriceRate	Int	N	Price rate
DST_id	varchar (100)	N	Datastream ID supplied by provider

3.2.4.3. MEMORY OBJECTS: ORDER FORM

The most important object managed by the e-Commerce Engine is the "Order Form". This object will hold all the information about the products that the user is purchasing and the relative shipping and billing addresses.

In particular, it is filled with data from table "_basket", and is then completed according to the various choices performed by the user when he purchases a product (i.e. shipping methods, supply mediums, addresses, etc.) and by the pipelines.

The Order Form is then stored to the "_Receipt" table when the user finalises the purchase. Note that fields beginning with an underscore sign ("_") are not saved to database tables (i.e. user's critical information)

The Figure 5 below provides a detailed analysis of the Order Form object:

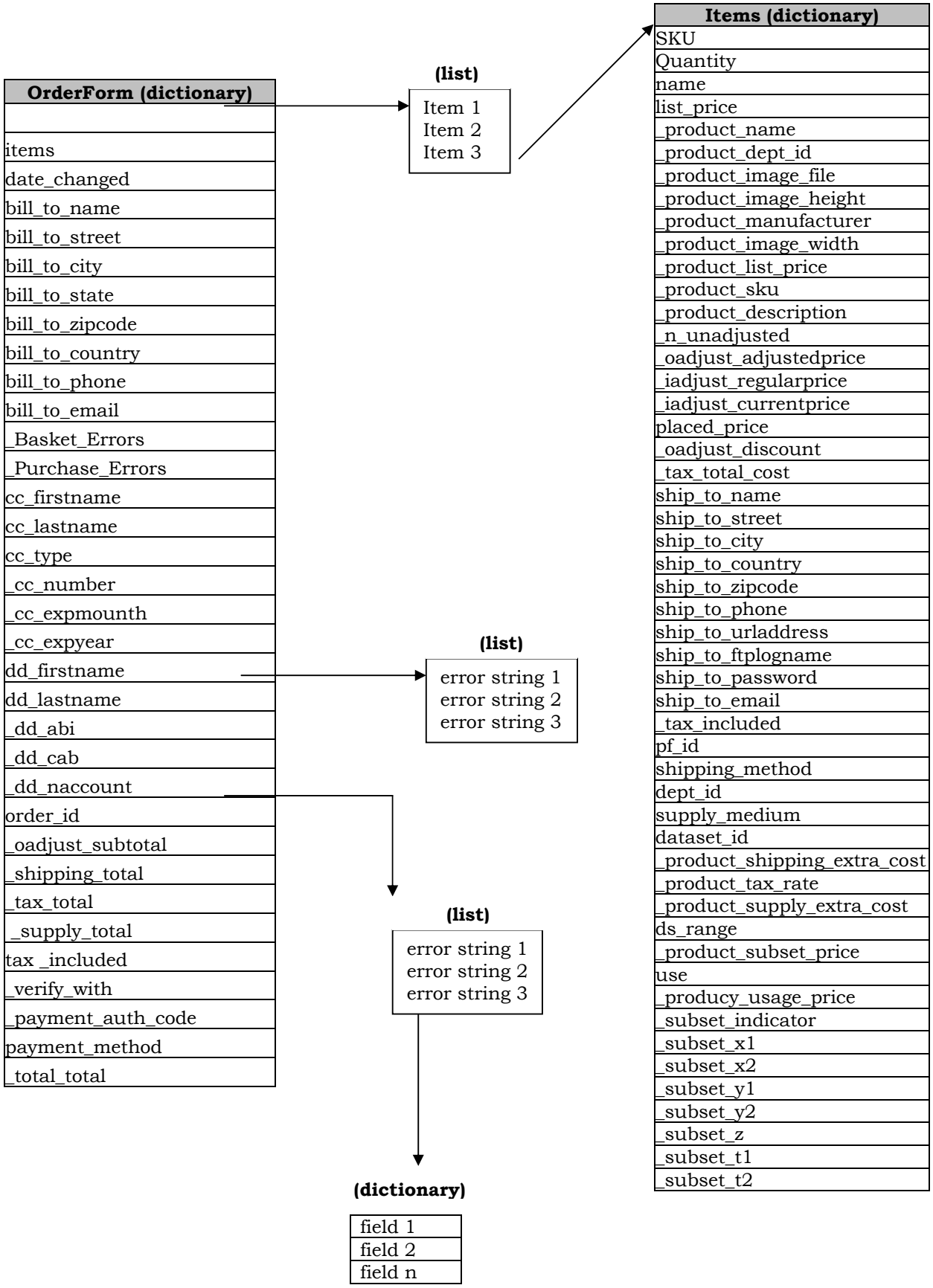


Figure 5 - Order Form Object Structure

A brief description of each element seen in the diagram is provided below:

Order Form Level Fields

items is a pointer to a list of items contained in the Order Form (see below)

date_changed is the Order Form date of change

bill_to_name
bill_to_street
bill_to_city
bill_to_state
bill_to_zipcode
bill_to_country
bill_to_phone
bill_to_email

} represent the billing information

_Basket_Errors points to a list errors in the Shopping Basket

_Purchase_Errors points to a list of purchase errors

_cc_firstname
_cc_last_name
_cc_type
_cc_number
_cc_expmonth
_cc_expyear

} represent the credit card information

order_id is the receipt number, and represents the unique number to identify the receipt and the order

_oadjust_subtotal is the item._oadjust_adjustedprice sum, that is the sum of partial total

_shipping_total is the sum of the shipping costs

_tax_total is the sum of tax costs

_supply_total is the sum of the supply costs

_tax_included is always equal at zero

_verify_with it's used to check that when passing from a page to another one the fields specified are not changed

payment auth_code

payment_method represents the name of the payment method that the customer chooses

total_total is the sum that the customer must pay.

Item Level Fields:

SKU identifier string, this is a sub-identifier for the product

Quantity is the number of the same items that the customer wants to purchase

name product name

list_price is the price of single item (not discounted)

_product_name
_product_dept_id
_product_image_file
_product_image_height
_product_width
_product_manufacturer
_product_list_price
_product_sku
_product_description

represent the product information (see table "_product")

_n_unadjusted is initialised by the pipeline, with promotion value if there is a promotion

_oadjust_adjustedprice is initialised by the pipeline with the discounted price (individual or category discount)

_iadjust_regularprice is initialised by the pipeline, in base of "list_price" field for each item in item list

_iadjust_currentprice is the sale price

placed_price final price

_oadjust_discount is the individual or category discount amount

ship_to_name
ship_to_street
ship_to_city
ship_to_country
ship_to_zipcode
ship_to_phone
ship_to_urladdress
ship_to_ftplogname
ship_to_password
ship_to_email

represent the shipping information

_tax_total_cost is the tax costs for each item

_tax_included is always equal to zero

pf_id is the identifier number for the product, created by MERMAID

dept_id is the data provider identifier code

shipping_method is the shipping method chosen by the customer

supply_medium is the supply medium chosen by the customer

dataset_id is the identifier string for the product used by providers at their site

_product_shipping_extra_cost
_product_supply_extra_cost
_product_tax_rate
_product_subset_price
_product_usage_price
} represent the extra costs information

ds_range is the product share that customer want to purchase

use is the usage category that the customer choose when purchase the product

_subset_indicator indicates whether the product is a subset of a bigger dataset or not
_subset_x1
_subset_x2
_subset_y1
_subset_y2
_subset_z
_subset_t1
_subset_t2
} subset coordinates:
('x' start/end, 'y' start/end, selected 'z', time start/end)

3.2.4.4. PIPELINES CUSTOMISATION

Either the "Purchase" pipeline or the "Plan" pipeline will have been changed.
In particular, the "Plan" pipeline, used to finalise and validate the Order Form requires the following:

- new component to calculate shipping costs at item level
- new component to calculate supply costs at item level
- new component to calculate taxes costs at item level

The "Purchase" pipeline, used at the order completion and payment time requires:

- interface with e-payment services
- store more information into receipts

3.2.4.5. ASP PAGES

Although MSSCE provides wizards to create site pages, most of them have been modified and many new ones created. In general it is difficult to distingusih between e-Commerce pages and MERMAID pages, as many pages may activate more than MERMAID component, including the e-Commerce Engine. Therefore, the creation of such pages is not really dealt with the e-Commerce Engine, it is rather dealt with the system integration WorkPackage (WP6).

However, a general overview of the MERMAID pages is shown below:

User homepages: these are providers & consumers lobbies and profile update pages. e-Commerce is partially involved here, as user profiling pages are stored in the e-Commerce DB. WP6 also deals with these pages.

Product Search: this pages involve both the CatalogueEngine (for the metadata) and the eCommerce (for the pricing of the search result).

Shopping Basket: list of products the user is about to purchase. e-Commerce only

Purchasing details: shipping, supply medium details. e-Commerce only

Payment: billing details and transaction results. e-Commerce only

Product Delivery: after payment, the DataAccessEngine may be activated to send the dataset to the customer

Product metadata edit: involving the CatalogueEngine

Product pricing edit: including all the various discounts: e-Commerce stuff

Product upload: DataAccessEngine responsibility

3.2.5. E-Payment Integration

A list of the possible components that must be integrated with MS Site Server to perform on-line payments is provided in the appendix of this document.

Briefly, the customisation of the software is:

- payment methods supported
- security standards
- setup and running costs

The general idea behind these services is common and it is to offer a client-side component able to connect to a remote server (through some secure communication line) and passing to it the payment details (i.e. credit card number, bank account number, etc.). The server is then responsible to connect to the required service (i.e. Visa / Mastercard circuits, banks, etc.) and therefore perform the transaction of moving the money from the customer site to a bank specified by the merchant when he signs the contract with this service. Figure 6 provides a diagram to explain the situation:

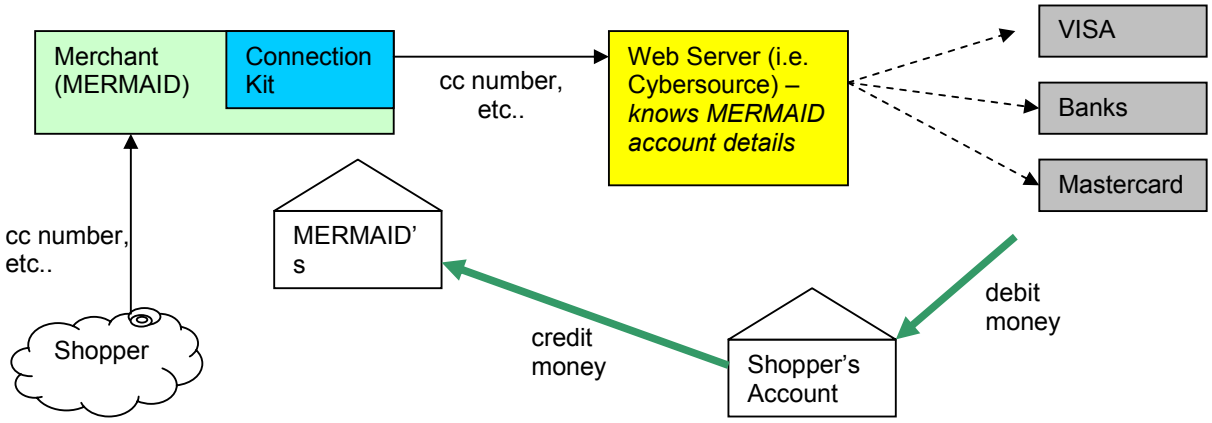


Figure 6 : E-Payment software Integration

Using such services, MERMAID does not have to store any shopper's vital information (i.e. credit card numbers, etc.) as these values are just passed (encrypted) 'on-the-fly' from the web page the shopper is filling to the WebServer providing the service. Of course, this service is responsible then for the safety of this information.

What is required at the merchant's site (MERMAID site) is to integrate the connection kit. In the most fortuitous cases, such web services vendors are able to offer an SSCE pipeline component acting as a connection kit: this would be the best solution, as a complete integration with the SSCE business processes is performed, and the power of SSCE transacted pipelines can be fully deployed.

The appendix details which vendors are offering a component for SSCE rather than a generic connection kit.

3.2.6. Integration into MERMAID architecture issues

As previously explained, in order to integrate the e-Commerce Engine with the other components, it is necessary to define a unified access point to all the functionality offered.

This access point, called “e-Commerce Gateway” is a set of COM DLL objects, whose methods can be used by other components. Figure 7 illustrates a typical situation (for more details about the MERMAID broker architecture refer to the WP3 deliverable; MERMAID System High Level Design – D2).

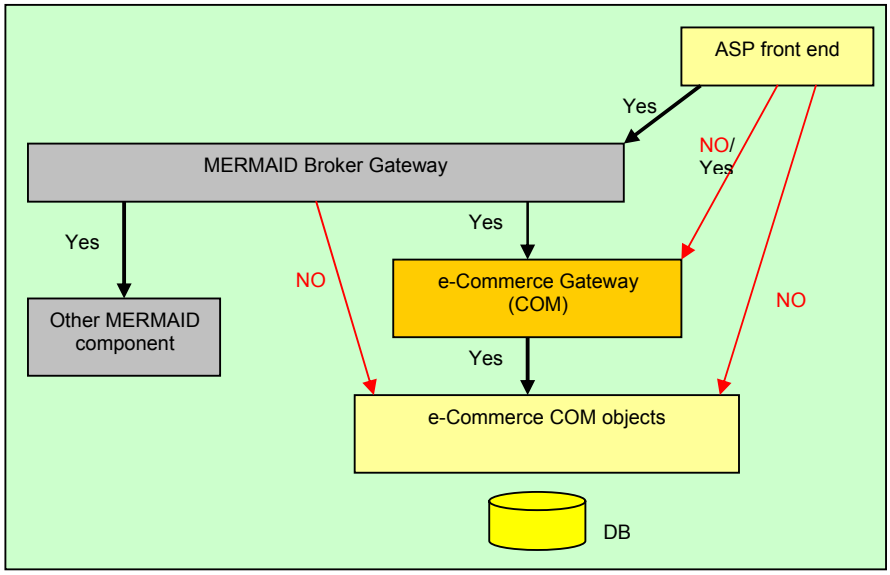


Figure 7 : Integration of e-Commerce Gateway and Engine.

It should be noted that all the components shown in Figure 7 will be running on the same machine. Remote connections to distributed components (i.e. the Data Access Engine) will be handled by the “MERMAID Broker Gateway”.

Of course, there is no way to forbid another application to access SSCE objects directly, but this could cause incoherent or wrong behaviours (red arrows in the diagrams should not be allowed or implemented).

Therefore, to use the functionality of the e-Commerce Engine it is necessary to go through the e-Commerce Gateway.

ASP pages should refer to MERMAID Broker Gateway objects, as one page may be the interface to a process that involves more than one component. Of course there could be cases in which one ASP page is only used to perform e-Commerce operations, and in this case, the MERMAID Broker method supporting such a page will simply be a one-to-one mapping of a method offered by the e-Commerce Gateway. An alternative would be to define those processes that do not involve other MERMAID components and let ASP pages make direct use of the e-Commerce Gateway only for them.

Tables 3 and 4 provide an **overview** of the functionality offered by the e-Commerce Gateway:

Data Consumers functions:

Function Name	Description	IN parameters	OUT Parameters
login consumer	authenticate a consumer, create into memory the shopping basket structure	user name, password	user ID (relative to session)
get product price	including discounts based on user's category, dataset usage and subset	product ID, user ID, subset percentage, usage	product price
add product to basket	add a product to the user's shopping basket	product ID, user ID	ok or error code
get product shipping methods and costs	get the list of possible shipping methods and shipping costs for a given product, with delivery times	product ID	method, cost and delivery time list
get product supply mediums and costs	get the list of possible supply mediums and supply costs for a given product	product ID	medium and cost list
process order	prepare the shopping basket to be purchased, providing the last details about shipping, supply medium. Perform prices computation and information completeness verification by the running of the "process" pipeline	user ID, list of: product ID, shipping method names, shipping address (according to shipping method) and supply medium names (for each item of the shopping basket)	ok or error code
purchase order	finalise order purchase, by specifying billing address and payment details (this implies the run of a "purchase" pipeline) This function must be called only after a successful run of the "process order " function. In case of error in the "purchase order" function it is not necessary to run again the "process order" function	billing address, payment method, payment details	ok or error code
get receipts id list	list of user's receipts ID	user ID	list of receipts ID
get receipt	returns a specific receipt	user ID, receipt ID	document describing receipt

Table 3: e-Commerce Gateway functions for Data Consumers

Data Providers functions:

Function Name	Description	IN parameters	OUT Parameters
login provider	authenticate a provider	user name, password	provider ID (relative to session)
create product	create a product in the e-commerce DB. Product ID should come from another component	product ID	ok or error code
add / update / delete product shipping method	change name and price relative to a shipping method for a given product	provider ID, product ID, shipping method name (a,u,d), shipping method cost (a,u)	ok or error code
add / update / delete product supply medium	change name and price relative to a supply medium for a given product	provider ID, product ID, supply medium name (a,u,d), supply cost (a,u)	ok or error code
add / update / delete product subset discount	change discounts values for subset for a given product	provider ID, product ID, subset low value (a,u,d), subset high value (a,u), discount rate (a,u)	ok or error code
add / update / delete product consumer's category	change discounts values for categories for a given product	provider ID, product ID, category name (a,u,d), discount rate (a,u)	ok or error code
add / update / delete product usage discount	change discounts values for usage for a given product	provider ID, product ID, usage name (a,u,d), discount rate (a,u)	ok or error code
get discount table	retrieves a discount table (category, usage, subset) for a given product	provider ID, product ID, discount type	discount list
purchase datawarehouse space	purchase MERMAID datawarehouse space for a given period	provider ID, amount required, payment method, payment details, billing address	ok or error code
get datawarehouse space	retrieves either the total and available MERMAID datawarehouse space available for a given provider	provider ID	datawarehouse space total and available

Table 4: e-Commerce Gateway functions for Data Providers

3.3. Final architecture of the e-Commerce Engine

Figure 8 summarises all the considerations previously discussed and gives a global view of the component that will build the e-Commerce Engine:

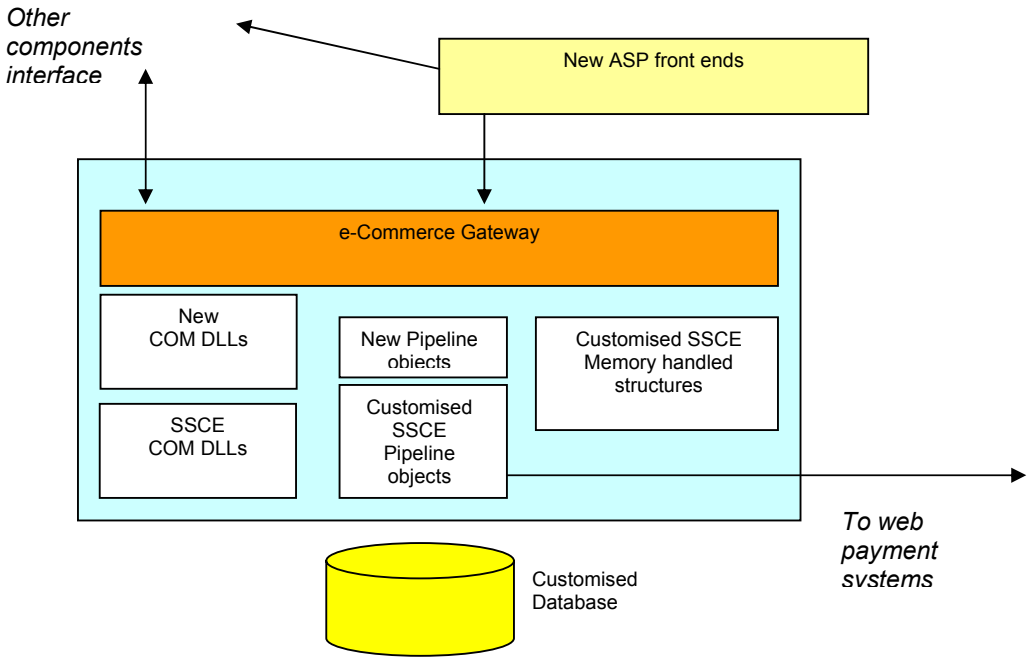


Figure 8: Diagram showing e-Commerce Engine components and interface

The architecture is similar to the SSCE one, but has these new components:

- new COM DLLS to implement the new features required
- e-Commerce Gateway to handle connections/integration with other components
- new ASP pages

In addition, the following existing components have been customised to change their behavior:

- memory objects (i.e. the new order form)
- pipelines: pipelines existing components have been parameterised, new components created and possible third party component (e-payment connection) tested

A. Appendix A: e-Payment Web Solutions

The objective of this appendix is to provide a quick overview of web payment solutions, and to understand how the MERMAID e-Commerce Engine could be integrated into such services. For the moment the purpose is not to choose the best one (that may also change in time and in different countries) but rather to understand what can and cannot be done.

A more detailed analysis is therefore required at the time of exploitation, when it will be necessary to take a concrete decision. Also, by that time, it will be possible that new payment methods, new solution providers or new software could be released, offering new business possibilities.

A.1. e-Payment Methods

There are many vendors offering solutions for on-line payment. There are various payment methods that can be used to perform the on-line payment, that are summarised here. It must be noted that the only method that is universally accepted, and thus not dependent on specific country laws, and does not require shoppers (or merchants) to have an account on a specific bank, is still the credit card.

A.1.1 Credit Cards

Credit cards are probably the most popular way to make a non-cash payment. They are also the most accepted payment method over the Internet. Credit cards can include corporate cards, special collective cards that can be configured to be authorised in the purchasing of only some products (useful therefore in the enterprise office staff purchasing for example).

A.1.2. e-Checks

e-Checks are similar to normal cheques, except the fact that they are processed electronically, in an encrypted way, usually through the use of software capable of encrypting information and connecting to e-check processing systems. Both shopper and merchant need to have an e-check enabled bank account. In general, even if quicker than normal checks to be processed, e-checks still require 1 to 3 days, and for this reason are not suitable for shopping of those products that must be delivered on-line.

Links:
www.netchex.com
www.mkn.co.uk/
www.fstc.org/

A.1.3. e-Cash

e-Cash is nothing more than strings of bits representing money. First it is necessary for the consumer to have an e-Cash enabled bank account, then, through software provided by the bank it is possible to transfer some money from the traditional account to their PC. Then they can use the “virtual money” to pay to sites accepting e-cash.

Links:
www.magnacash.com
www.digicash.com
www.mondex.com

A.1.4. Smart Cards

Smart cards are similar to credit cards: they have a microchip on board and can be uploaded with money directly from automated banking machines

A.2. e-Payment solutions over the Web analysis

Below is a brief analysis of some web sites offering e-payment solutions. Those offering MS SiteServer pipelines component have been preferred. Note that many data are still incomplete, due to difficulties into getting information from these sites (especially about pricing).

A.2.1. CyberCash

www.cybercash.com

CyberCash offers a broad set of tools for solution ranging from web payment to physical stores payment, amongst which:

CashRegister component for MS SiteServer	
Descriptions:	SSCE Pipeline component connecting to the CyberCash payment server http://www.cybercash.com/cashregister/
Architecture:	COM/pipeline
Payment Methods:	credit card
Cost:	Setup Fee: 495 $ Monthly Fee: 20 $ Fee per transaction 0.20 $
Currency:	Accepts all kind of currencies
Notes:	tested successfully

CashRegister connection Kit	
Descriptions:	Similar to the SSCE Pipeline component but enabling more integration technologies http://www.cybercash.com/cashregister/
Architecture:	C API, ASP pages, Perl
Payment Methods:	credit cards, e-checks (ACH)
Cost:	Setup Fee: 495 $ Monthly Fee: 20 $ Fee per transaction 0.20 $
Currency:	Accepts all kind of currencies
Notes:	

A.2.2 VeriSign

www.verisign.com

Versign offers a products suite to support all web payment and security needs as well as wireless devices, registry services, etc.

Payflow PRO	
Descriptions:	SSCE Pipeline component connecting to the CyberCash payment server http://www.verisign.com/products/payflow/pro/index.html
Architecture:	ASP, COM-DLL, C library, Java, Perl, MS Site Server
Payment Methods:	credit cards, e-checks (ACH)
Cost:	For less than 5000 monthly transactions: Setup Fee: 250$ Monthly Fee: 60 $ For more than 5000 monthly transactions: 995 $
Currency:	Accepts only transaction in US DOLLARS !!!
Notes:	SSCE version tested succesfully

A.2.3. CyberSource

www.cybersource.com

Cybersource offers either products and services for commerce infrastructure

CyberSource Commerce Component	
Descriptions:	SSCE Pipeline component connecting to the CyberSource payment server. The components is also performing many other services such as addresses verification, certificates release, fraud controls, tax calculation, etc… http://cybersource.com/services/payment/
Architecture:	COM/pipeline
Payment Methods:	credit cards, e-checks, MagnaCash
Cost:	n/a
Currency:	all currencies for credit cards, US dollars and Canadian dollars for the other methods
Notes:	tested successfully with credit cards

A.2.4. Banca Sella

www.bancasella.it/ecommerce.html

Banca Sella is an Italian bank offering e-commerce services. Recently released the first off-the-shelf product called “GestPay”

GestPay	
Descriptions:	Windows utilities to handle connection to GestPay site. Payment page can be either on client’s site either on GestPay site http://ecommerce.sella.it/b2c/gestpay_index.htm
Architecture:	Java
Payment Methods:	credit cards, direct debit on BancaSella account
Cost:	Different versions available: Basic: about 8 $ / month Advanced: about 15 $ month Professional: about 20 $ / month Transaction Fee: specific to every vendor – no information available !!! Activation cost: 100 $
Currency:	Italian lira and EURO
Notes:	direct debit only works if both shopper and merchant have an account on BancaSella No demo available

A.2.5. ClearCommerce

www.clearcommerce.com

ClearCommerce offers products for Merchants, B2B commerce and Service Providers

ClearLink component for MS SiteServer	
Descriptions:	SSCE Pipeline component connecting to the ClearCommerce payment server. Also perform shipping address verification and taxes computation services and fraud protection http://www.clearcommerce.com/cs2000/
Architecture:	COM/pipeline
Payment Methods:	credit cards
Cost:	n/a
Currency:	not specified – only US dollars ?
Notes:	not tested

A.2.6. Swift

www.swift.com

SWIFT is an industry owned co-operative supplying secure messaging services and interface software to over 7,000 financial institutions in 192 countries.

ePayment Plus	
Descriptions:	end to end payments for global e-Commerce http://www.swift.com/index.cfm?item_id=3100
Architecture:	n/a
Payment Methods:	- should interface to banks directly -
Cost:	n/a
Currency:	not specified – EURO at lease hopefully
Notes:	1st release scheduled by 2nd quarter 2001

A.2.7. Costs Comparative Table

For those vendors publishing their prices, it's been possible to perform a quick cost comparative analysis, just to get an idea of them

Product:	Cost of 1 year with 100 monthly transactions:	Cost of 1 year with 1000 monthly transactions:	Cost of 1 year with 10.000 monthly tranactions:
CyberCash - CashRegister	975 $ (1st year), 480 $	3135 $ (1st year), 2640 $	24,735$ (1st year), 24,240 $
VeriSign – PayFlow PRO	970 $ (1st year), 720 $	970 $ (1st year), 720 $	995 $
Banca Sella – GestPay Basic	196 $ + 100*x (1st year), 96 $ + 100*x	196 $ + 1000*x (1st year), 96 $ + 1000*x	196 $ + 1000*x (1st year), 96 $ + 1000*x