



Sizing and Deployment

Informatica MDM - Product 360

Version: 8.1.1

1 Table of Contents

1	Table of Contents	2
2	Objective	5
3	Module Overview.....	5
3.1	Application Server	5
3.2	Control Center	6
3.3	Desktop Client	6
3.4	Web and REST Clients	6
3.5	Fulltext Search Index.....	6
3.6	Audit Trail	6
3.7	Informatica BPM.....	6
3.8	Physical File Share.....	6
3.9	Database	6
4	Resource Limitations.....	6
4.1	CPU and CPU Cores	7
4.2	Memory.....	7
4.2.1	Multi-Server considerations	7
4.2.2	Caches	7
	Status Cache	7
	ListModel Cache	7
	ListProcessor Cache.....	7
	DetailModel / DataGraph Cache	7
4.3	Network	8
5	Scaling Factors	8
5.1	Users	8
5.1.1	Power Users	8
5.1.2	Occasional Users	8
5.2	Items	8
5.3	Jobs	8

5.3.1	Import	8
	Phase 1	9
	Phase 2	9
5.3.2	Merge.....	9
5.3.3	Export.....	9
5.3.4	Data Quality	9
5.4	3rd Party	9
6	Logical Deployment	9
6.1	Single Server.....	9
6.2	Separate Jobs from Clients.....	10
6.3	Scale Out	11
7	Physical Deployment	11
7.1	Roles	11
7.2	Typical Deployment	12
7.3	High Availability with Load Balancing	12
7.3.1	Application Server with User Role.....	14
	Load Balancing	14
	Fail-over	14
7.3.2	Application Server with Job Role	14
	Load Balancing	14
	Fail over	14
7.3.3	Application Server with Web or Service API Clients	14
	Load Balancing	14
	Fail-over	14
7.3.4	Media Manager File Server.....	15
	Load Balancing / Fail over	15
7.3.5	Media Manager Web Server	15
	Load Balancing	15
	Fail-over	15
7.3.6	Media Manager Server.....	15
7.3.7	Audit Trail Server.....	15
	Load Balancing	15
	Fail-over	15
7.3.8	Audit Trail Database	15

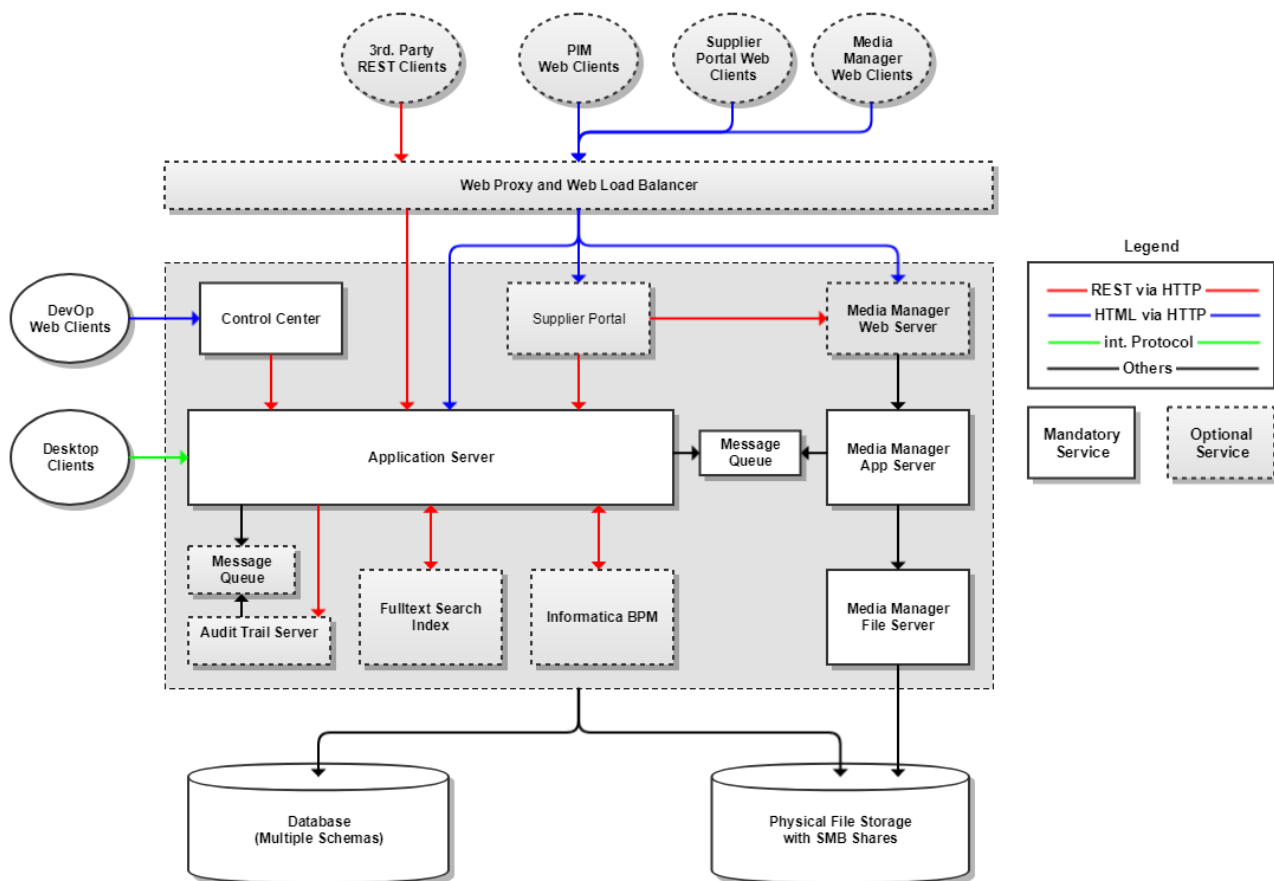
7.3.9	Database Server	15
7.3.10	Supplier Portal Server	16
	Load Balancing	16
	Fail-over	16
	Configuration.....	16
7.3.11	Informatica BPM Server	16
	Fail-over	16
7.3.12	Message Queue Server.....	16
7.4	High Availability without Load Balancing	16
7.4.1	Application Server Cluster	17
7.4.2	Media Manager, Audit Trail and Supplier Portal	17
7.4.3	Media Manager File Server.....	17
7.4.4	Database Server	17
8	Limitations.....	18
8.1	Number of Servers in the PIM Server cluster	18
8.2	Database Performance.....	18
9	Sizing Calculator	18
9.1	How does it work?	18
9.2	The Excel Sheets.....	19
9.2.1	Introduction	19
9.2.2	Questionnaire	19
	Objects.....	19
	Users.....	19
	Jobs	20
	Digital Assets	20
	Modules.....	20
	Printable Report	21
	Number of Cores.....	21
	Memory and Disk space	21
	Database Server.....	21
	Database Storage	21

The sizing guide provides general knowledge on the sizing criteria of the Informatica MDM - Product 360 application. It contains an overview on the available modules, the needed resources and the deployment possibilities. To get a general idea on the needed resources for a specific customer scenario we provide a [Sizing Calculator](#).

2 Objective

Informatica MDM - Product 360 is the central product information platform for e-commerce and multi-channel commerce. The solution provides distributors and manufacturers with a central solution for all product data in a central data source for the purpose of distribution in all communication channels and languages. Key elements of the solution include the import (Onboarding) and central mastering of highly structured product data and the corresponding media assets. This means that centrally managed product data can be adapted to suit the needs of various target systems (multi-channel), such as online shops or print systems, when it is exported in data formats based on CSV or XML. Product 360 comprises the following three functional areas along the supply chain from on-boarding of product data to the multi-channel export into various target systems.

3 Module Overview



3.1 Application Server

The application server of Informatica MDM - Product 360 is the highly extensible core of the application. It provides all product data management functionalities for different clients and executes the majority of the load of the system. It performs imports, merges, exports as well as other mass operations like search & replace. It is highly optimized for multi-core systems,

provides several caching technologies to reduce the load on the database and provides a powerful REST based api which can be used to integrate Product 360 with other 3rd party applications.

3.2 Control Center

The control center is a small, lightweight application who is able to control and manage the application server. It is especially useful in a multi-server deployment scenario where managing multiple application servers might be cumbersome. It provides general management functionalities like start and stop of servers as well as typical operational functions. The control center is a separate application and an own service in the host machine. It's must be installed on each host which has an application server running. Additionally to the build-in management functionalities it also provides access to it's data by the SNMP protocol.

3.3 Desktop Client

The Informatica MDM - Product 360 Desktop Client is a desktop application for Windows. It provides a very flexible UI and a powerful editing interface which rich mass data editing functionalities. The Desktop Client is directly connected to one of the application servers using a proprietary protocol which is implemented directly on the low-level TCP/IP sockets from Java. The Desktop Client is designed and implemented as intranet application and needs a reasonable amount of bandwidth. Mass data handling after all transmits a lot of data.

3.4 Web and REST Clients

All HTTP based clients should be connected to a reverse proxy for security purposes. This proxy can also provide a load balancing logic in case of a multi-server deployment. Please make sure that the load balancer will forward requests from the same client always to the same server (session stickiness).

3.5 Fulltext Search Index

The web search module provides an full text index service which is currently accessible by the web client only. The web search module is optional and not required in case the functionality is not needed.

3.6 Audit Trail

The audit trail module stores all changes anyone or anything does to the Product 360 objects in a separate database schema. The Audit Trail server is an own service and receives the change information using the Message Queue service. Audit Trail records are written asynchronously to the audit trail schema.

3.7 Informatica BPM

The Informatica BPM "module" in the picture above is actually the full Informatica BPM product which is integrated using REST Api calls to and from the Product 360 application server.

3.8 Physical File Share

All servers of the Informatica MDM - Product 360 deployment need access to the same physical file share in order to be able to distribute their workload inside the application server cluster. This file share must be provided by a Windows or Linux host which is also capable of

3.9 Database

All application server are directly connected to the database. Multiple schemas are needed in the database. The database sizing should be done according to the specifications the database vendors provide. In general Product 360 is a OLTP application with a typical workload for these kinds of applications.

4 Resource Limitations

A short overview on the resources the Product 360 application relies on.

4.1 CPU and CPU Cores

Most prominent resource these days are of course the CPU or CPU cores. The number of total cores in the application server more or less directly maps to the performance and maximum load capacity of the application. The Product 360 application server is optimized for multi-core systems in such way that mass operations are distributed in multiple parallel worker threads which are then processed by the available cores. In general this is true for all typical mass operations like import, merge, search & replace, data quality but also other operations in which a user can select multiple objects like classification, attribute maintenance and more.

So, the needed number of cores is bound to the number of concurrent users and jobs and how many objects those users and jobs are processing at the time. It's not really relevant how many objects are in the database in total.

4.2 Memory

The memory consumption, in contrary to the CPUs is not so much dependent on the number of users or jobs but on the total number of objects in the database. For large exports or lists a significant amount of memory is needed for the time the model is processed which makes it important to have enough memory resources available for those tasks. The number of concurrent users is not so important since the application server holds only very little data user specific.

4.2.1 Multi-Server considerations

As with Version 8 we do not yet share caches across multiple servers. So the memory needs are identical for all servers, no matter how many servers are used.

4.2.2 Caches

Product 360 has various caches in place which store data in memory in order to reduce the number of requests to the database. For the memory requirements it's important to be aware that some of those caches are held completely in memory. In order to get a better impression we list a few of them here:

Status Cache

The status cache holds current results of all data quality checks for all objects which have such results in memory. This is needed in order to be able to filter lists of items very fast for these data quality results. The memory footprint is not as big as one would expect as those results are stored as bit sets. Since the application relies on the availability of this cache and it's completeness, you can not adjust the number of elements in this cache.

ListModel Cache

The Service API has the possibility to cache complete list models. This is useful for paged list access from Service API clients so they don't need to re-evaluate the query for every page. In case the Service API is not used correctly you might run into memory issues in case this cache is filled with really big list models. The maximum number of cached list model is configured in the `ehcache.xml` file.

ListProcessor Cache

For analysis purposes needed for example in the dashboards, we do have a ListProcessor cache. This cache holds all records for the requested fields in memory and is able to execute analysis logic on those fields. Mainly used by the dashboard it's size is directly bound to the number of dashboards and how many fields are used in those.

DetailModel / DataGraph Cache


Detail models might hold all data for a single object. It's in the nature of those "big" objects to need more time to load from the persistence storage. That's why the detail model cache comes into place. Multiple requests for the same objects will be handled by this cache. Since detail models are quite big from a memory footprint point of view, this cache will use all available memory which is not needed for other purposes.

Every load of a detail model will put the data into this cache, every save or delete operation will invalidate the cache again. Typically objects like hierarchies, units, users and others do profit the most from this cache - as they are not changed that often.

4.3 Network

Product 360 is optimized for mass data handling of rich product data. Using powerful grouping functions in tables with millions of records in multiple views is a central key feature of the desktop client. The Desktop Client is designed and implemented as an intranet application which typically has at least 100 MBit network bandwidth available. Although the Desktop Client works also with less, you might experience longer waiting times in case you load larger tables. The actually needed bandwidth is very different depending on the actual use cases of the users so this sizing guide can not give a definitive answer to this.

Table views in the web and desktop client, the export and also the service api work with large table like structure with possible millions of rows for a few tens of fields. In order to provide these table models in a fast way the network bandwidth between application server and database server must be as best as possible. At least 1 GBit, better would be 10 GBit is required.

 We do not recommend to have the database in a separate network or in the cloud. For cloud deployments you need to make sure that also the application servers are in the cloud and the network speed between application and database server is fast enough.

5 Scaling Factors

5.1 Users

When we talk about users there exist a lot of different definitions. For this sizing guide we will concentrate on two of them.

5.1.1 Power Users

The power user is a user which works nearly the whole working day within the Product 360 Application. Either with the web client or the desktop client. Typically we see that power users tend to use the desktop client thou. A power user uses mass data actions like search & replace, multi selection classification and cumulative attribute editing on a regular basis.

5.1.2 Occasional Users

An occasional user uses the Product 360 application maybe a few times a day till a few times a month. That's a big space of course and needs to be discussed during the sizing process with the customer. The default sizing calculator assumes that 4 occasional users produce as much load as one power user. Often this is not correct and needs to be adjusted since the reality for the customer is different.

5.2 Items

The number of items in combination with their complexity is even more important for the database than it is for the application server in Product 360. Handling 100 million items is an easy tasks if those items just have a few attributes. It's a different story in case the items have 200 attributes, multiple languages, hundreds of references to other objects and so on. Because of this all our benchmarks are performed with a defined set of objects, not only in number but also in complexity. Customers can compare the profiles with their needs and get a rough idea if they fit into that or exceed is drastically. Please see the Benchmark Profile Definition page for details on the tested complexity.

5.3 Jobs

Background jobs like imports, merges and exports are at least as important as the users in terms of scaling. All jobs which modify objects are implemented to use multi-threading in order to utilize as much processing power as available on the machine. The more CPU cores, the faster they finish, most times it's quite simple.

5.3.1 Import

The import is divided into two phases.

Phase 1

- Reading the import file
- Combining multiple rows for the same object
- Executing any Import Functions.

Phase 1 is single threaded and takes usually 10% of the overall execution time, depending on the used functions of course.

Phase 2

- Executing the business logic to create or update the objects

Phase 2 is multi-threaded and uses all available CPU cores to perform this. The more cores, the faster it gets.

5.3.2 Merge

The merge has a similar load profile than Phase 2 of the import. It's multi-threaded and reads the supplier catalog item and performs business logic to create/update the master catalog item.

5.3.3 Export

From most important jobs only the export is different in here since the export uses the mass data table like structures to process huge item counts. A single export utilizes a single core during this. So a job server with a lot of cores will not necessarily increase the export performance of a single export.

5.3.4 Data Quality

The data quality checks are usually executed automatically after an import, before a merge or before an export. It might also be scheduled regularly or manually through user interaction. The data quality job is also multi-threaded which means that the results are applied concurrently for all affected items.

As you can see, every job which creates or updates objects in Product 360 is implemented using multi-threading technology. If you have a lot of jobs running concurrently, it might be good to have one or multiple separate job servers. Unless that, the jobs will be either slowed down by the users, or vice versa. With Version 8 jobs do not distribute their own tasks on multiple servers. *So a single job will utilize only the CPU Cores of the server he is currently executed on.*

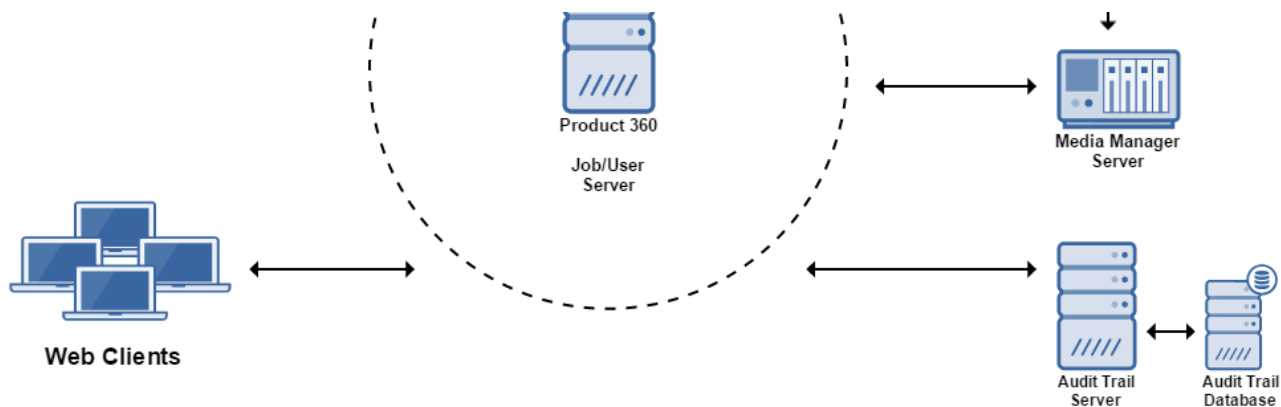
5.4 3rd Party

This category combines all 3rd party integration which do not use import or export jobs for the integration. Typically 3rd party applications use the Product 360 Service API to read and write object data, trigger jobs or query the meta data. In case a 3rd Party application sends a bigger list of objects which need to be modified, the Service API will also utilize multi-threading technology to perform this task as fast as possible. Because of this we usually recommend to have the Service API clients also use the job servers so they do not interfere with the desktop clients for the power users, but this needs to be discussed with the customer.

6 Logical Deployment

6.1 Single Server

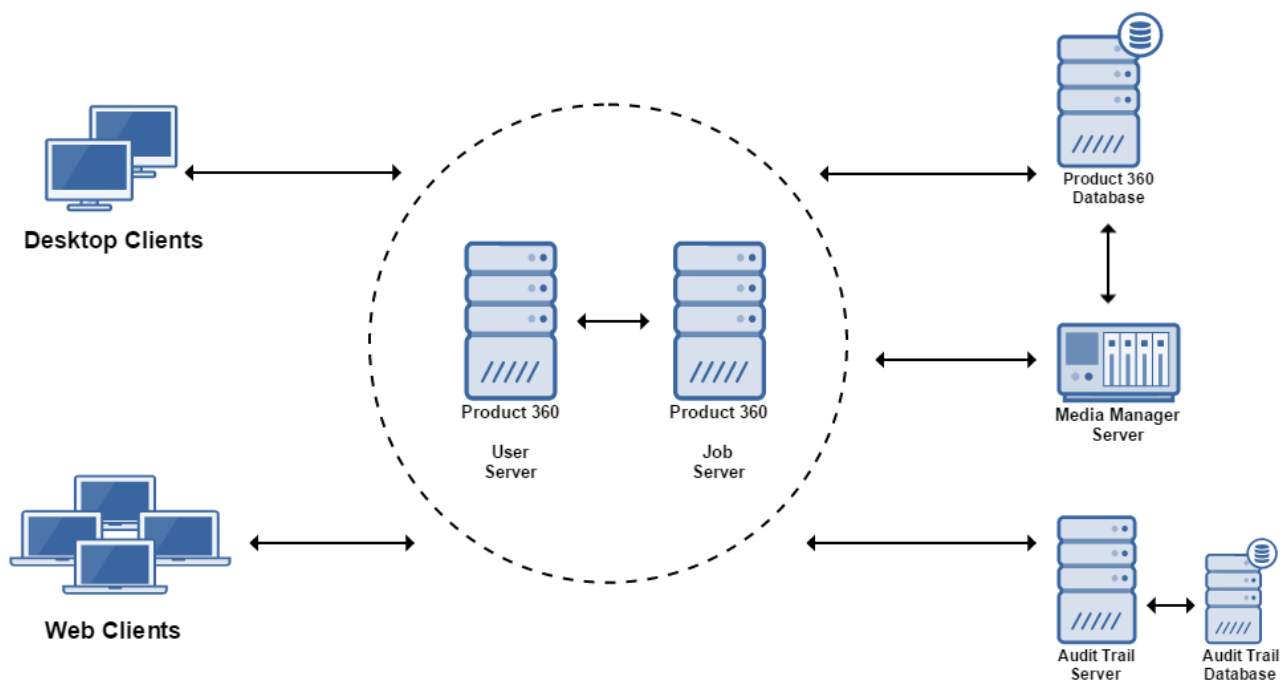




Long time, the single server deployment was the only one available. Which does not mean that all modules were installed on a single server thou. At least the database, the media manager, supplier portal and the audit trail servers could and have been installed on different hosts. However, the core application server which processes all client requests as well as the jobs was not there multiple times.

For small projects, or demo and training installations, the single server is still a valid option. It scales vertically with the number of cores. Also for scenarios in which jobs are only executed during non office business hours the single server deployment might be the right one.

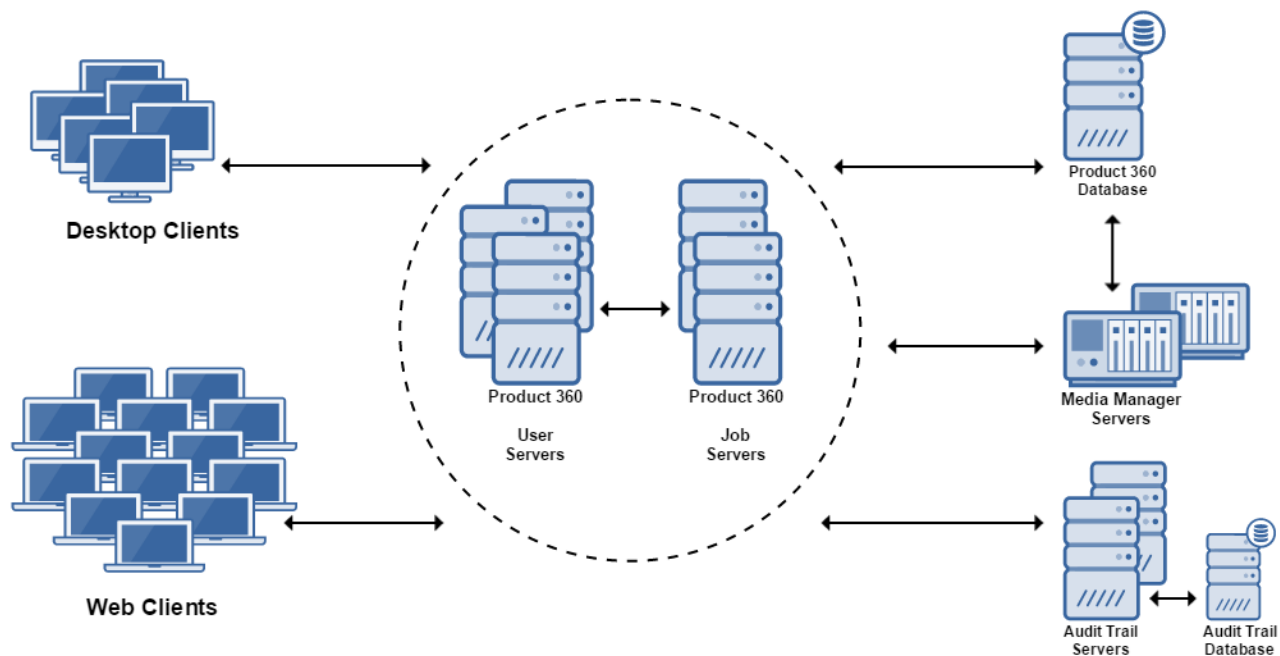
6.2 Separate Jobs from Clients



Depending on the number and type of background jobs and the time they need to be executed, a multi-server scenario is recommended. By separating the users from the jobs we make sure that response times for users don't degenerate, and job execution times also don't suffer from user load. Please note that the sizing calculator might return a relatively small number of cores for the jobs server - but usually it makes no sense to have less than 8 cores - or the jobs will finish not fast enough

from a single job perspective. However, it's totally feasible to route also the 3rd Party applications to the job server as well as BPM calls so that the job server has a fair share of work to do also when he's not under full power.

6.3 Scale Out



Based on the expected load every module and server can be scaled out horizontally. You can have multiple user and multiple job servers, as well as multiple audit trail servers and media manager or supplier portal servers. The physical deployment section will show in detail how this can look like.

7 Physical Deployment

7.1 Roles

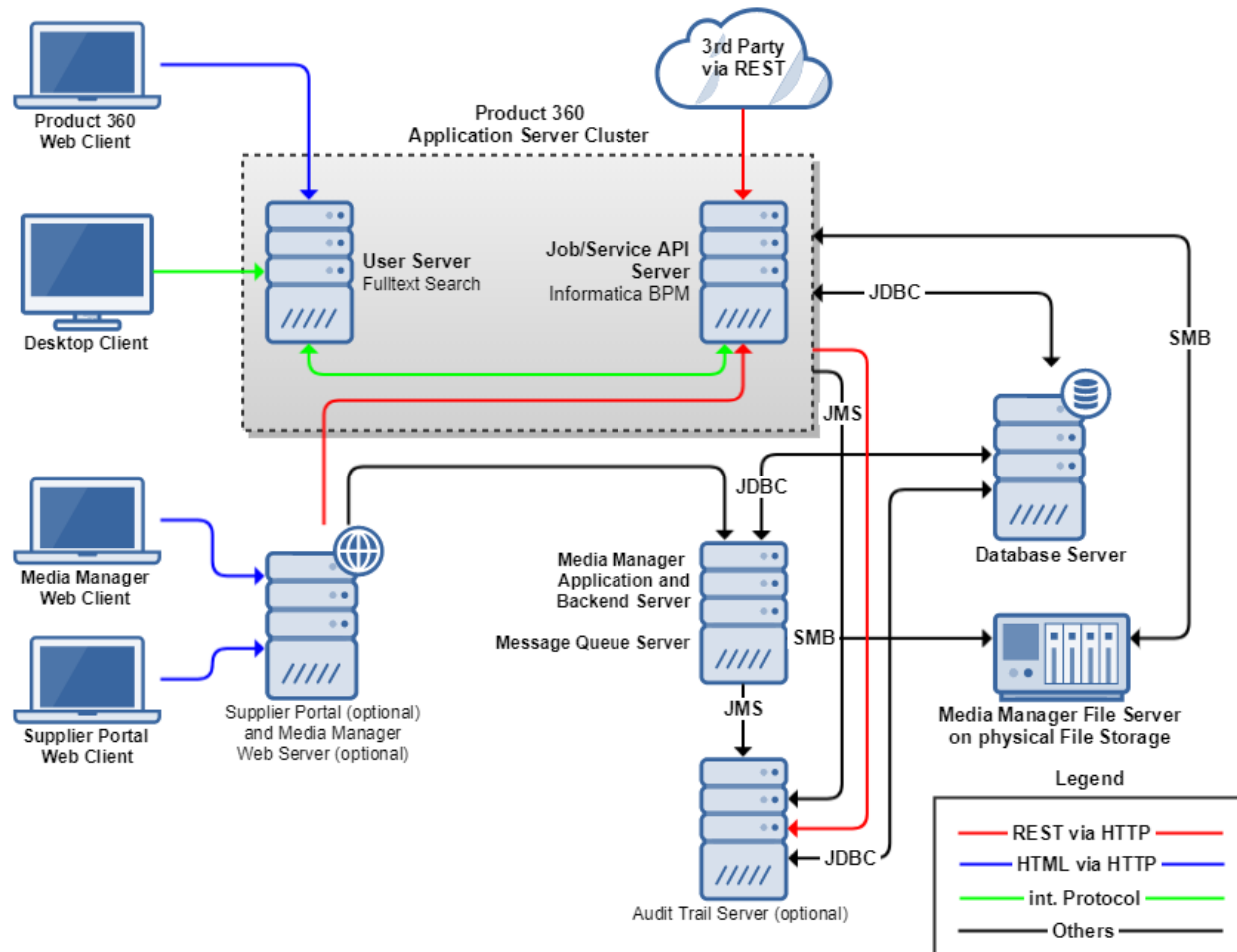
PIM servers are deployed absolutely identical. Thus, in theory, all of them can perform all operations. To be able to separate the type of load, we introduced server roles. Of course it is absolutely possible that a server has multiple, if not all roles.

- **CLIENT_SERVER**
The server is able to serve desktop clients. Desktop clients will be distributed across all servers which have this role
- **JOB_SERVER**
The server is able to execute background jobs. This includes user triggered jobs as well as scheduled jobs. So pretty much all jobs which can also be seen in the process overview. In case multiple servers have this role, the jobs will be distributed across them using a simple round robin algorithm.

Besides the Server roles which are configured in the server, the web/rest load balancing allows us to define two additional roles which need to be configured in the load balancer.

- **Web Clients**
This server only serves Product 360 Web Clients.
- **Rest Clients**
This server only serves Service API, or, REST clients (including the Supplier Portal)

7.2 Typical Deployment

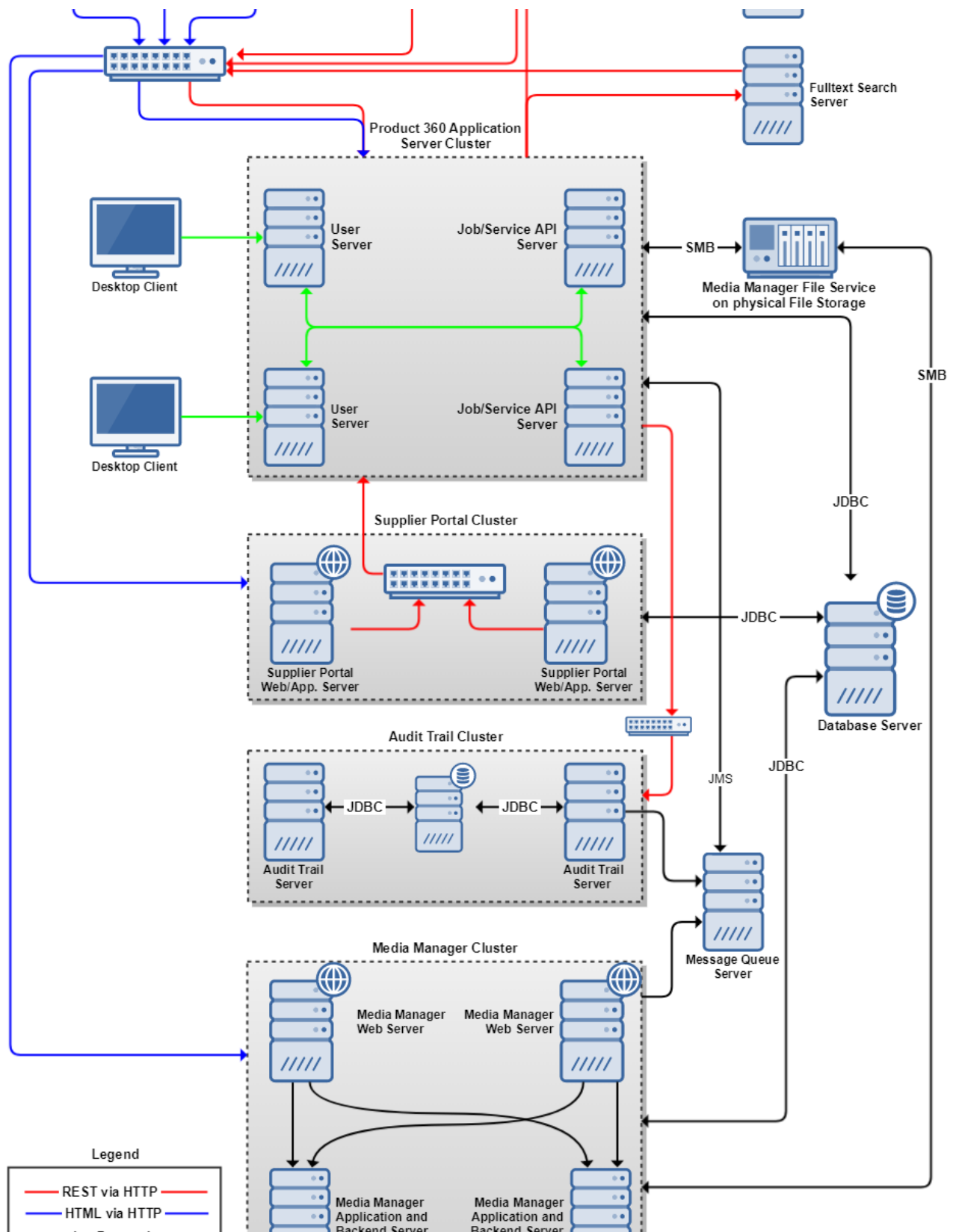


The typical deployment corresponds with the "Separate Jobs from Clients" scenario from above. In this picture you can see the communication paths from and to the modules of Product 360. Since this deployment does not have multiple servers for the web or rest clients, we don't need an HTTP load balancer. Optional modules like the Supplier Portal or Audit Trail Server can be combined on a single host in order to minimize the number hosts which need to be managed. Used Communication protocols include REST via HTTP, SMB for file transfer on the server side, as well as our own internal high-speed protocol for the communication between the servers and the desktop clients.

7.3 High Availability with Load Balancing

The so called "Full Blown Deployment" shows a scale out of every Product 360 module. It includes HTTP Load Balancer technology for all HTTP based clients (incl. REST). Modules are separated as much as feasible and scaled out individually. Please see this deployment as an example to what is possible with Product 360 Version 8 - not as a deployment which most users need.







7.3.1 Application Server with User Role

Load Balancing

Desktop clients automatically distribute them self on the available servers with the "CLIENT_SERVER" role. The used algorithm is based on the "least number of connected clients". At the very first startup of the client, a valid application server address is required. This can be just any server from the network, no matter which roles are configured. From this server, the client receives the hosts for all available client servers and automatically connects to the one with the least amount of connected clients.

Fail-over

In case the very last server with the "CLIENT_SERVER" role crashes, *all* other servers will automatically accept connection requests from clients. This build-in fail over logic is triggered automatically when the client reconnects to the network. Since the client remembers all servers of the network, he will try to connect to the very first one which is responsive to determine the current list of available servers to choose from.

7.3.2 Application Server with Job Role

Load Balancing

All servers with the "JOB_SERVER" role have a fully functional job framework up and running. Whenever a new job is scheduled the job information is written to the database. All Job servers are polling the job tables in regular intervals to see if new jobs are available which should be executed. The first server with a free Job Execution thread will pick the new job and executes it. It's not possible to define in advance that a specific server should execute a specific job. The distribution algorithm is round robin within the job servers. Internal system critical jobs are preferably executed on job servers.

Fail over

In case a Job server fails, all jobs he currently is executing are failed also and will be shown as cancelled. Users need to reschedule those jobs again manually in case it's not a regular job. For regular jobs, the next execution will be triggered as soon as it's time for it. A crash of the very last job server will trigger a startup of the internal system critical jobs on the client servers. User jobs can still be scheduled, but will not be executed until a job server is running again.

7.3.3 Application Server with Web or Service API Clients

For Web or Service API clients no special role configuration is needed on server side. Which server of the cluster serves Web Clients and which Service API clients must be configured within the Load Balancer. All servers of the network are, in general, capable to serve web as well as rest clients. Please see the Informatica MDM - Product 360 Configuration Manual for a configuration example.

Load Balancing

Web and Service API clients should be balanced using a standard HTTP Soft- or Hardware Load Balancer. We recommend to use the Apache Load Balancer but in general customers can use any state of the art Load Balancer which supports sticky sessions. The balancing algorithm to use can be decided by the customer depending on the capabilities of the Load Balancer.

Fail-over

In case the application server for web or rest clients crashes the clients need to reconnect, or re-execute the REST request which failed during execution. The HTTP Load Balancer will automatically direct the client to a still available server. It's recommended to configure the Load Balancer so it has a fall back server which is used only in case the primary servers are not responsive any more.

7.3.4 Media Manager File Server

Load Balancing / Fail over

The Media Manager File Server is a share of a Windows or Linux based server system. Please refer to an OS specific cluster solution. To prevent data loss use a professional backup solution or use at least a raid system for your shares.

7.3.5 Media Manager Web Server

Load Balancing

The Media Manager Web should be balanced using a standard HTTP soft- or hardware Load Balancer. We recommend to use the Apache Load Balancer but in general customers can use any state of the art Load Balancer which supports sticky sessions. The balancing algorithm to use can be decided by the customer depending on the capabilities of the Load Balancer.

Fail-over

In case the Media Manager web server crashes the clients need to reconnect. The HTTP Load Balancer will automatically direct the client to a still available server. It's recommended to configure the Load Balancer so it has a fall back server which is used only in case the primary servers are not responsive any more.

7.3.6 Media Manager Server

Load Balancing

Use horizontal installations (one Media Manager Server installation per machine) of the server modules to ensure a high work speed while increasing of the connections or jobs.

Fail over

Use horizontal installations (one Media Manager Server installation per machine) of the server modules. If one server is no longer available the other server(s) will take over the jobs.

7.3.7 Audit Trail Server

Load Balancing

The audit trail server is connected to the network using the message queue on the one hand. Therefor have multiple instances of the audit trail server up and running and configure them to use the same Message Queue Topic, which is, actually the default configuration. Each server will then individually poll the message queue for new messages and process them message by message. The Message Queue server takes care that not two servers process the same message. On the other hand, Application Servers access the Audit Trail Servers via REST for obtaining change information (which is then displayed in the Desktop Client. A load balancer is used to distribute the REST calls to all available Audit Trails Servers.

Fail-over

The audit trail server will notify the message queue as soon as it successfully has processed the message. In case it crashes before it did that, the message will automatically be picked up by the remaining audit trail servers making sure no message is lost. No special configuration is needed for this, just a minimum of two running audit trail servers

7.3.8 Audit Trail Database

Please refer to the deployment manuals for the Oracle or MS-SQL Server databases for high availability and fail-over deployment strategies.

7.3.9 Database Server

Please refer to the deployment manuals for the Oracle or MS-SQL Server databases for high availability and fail-over deployment strategies.

7.3.10 Supplier Portal Server

Load Balancing

The Supplier Portal is a web application and should be balanced using a standard HTTP soft- or hardware Load Balancer. We recommend to use the Apache Load Balancer but in general customers can use any state of the art Load Balancer which supports sticky sessions. The balancing algorithm to use can be decided by the customer depending on the capabilities of the Load Balancer.

Fail-over

In case the Supplier Portal server crashes the clients need to reconnect. The HTTP Load Balancer will automatically direct the client to a still available server. It's recommended to configure the Load Balancer so it has a fall back server which is used only in case the primary servers are not responsive any more.

Configuration

Most of the configuration options are shared between all Supplier Portal Server instances. However, there are some things to consider:

- All instances must have access to the same directory share that is used for file storage
- Background jobs for import job synchronization should be configured to run on one server only

Depending on the expected load, all internal http communication can be either routed to a single Product360 Application Server or load-balanced to multiple Application Servers. The same applies for the Media Manager Server. For details, check the Supplier Portal Configuration Guide.

7.3.11 Informatica BPM Server

Please refer to the Informatica BPM server documentation for details on how to set up a multi-server and high availability development.

Fail-over

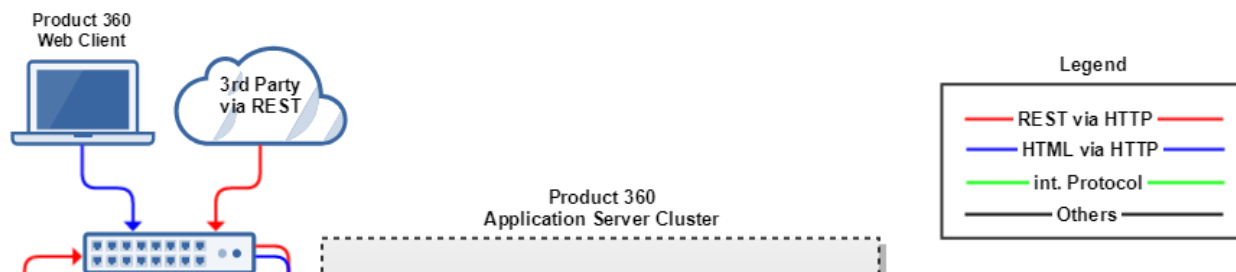
The MDM - Product 360 Application server will buffer all requests to the Workflow engine in case the BPM server is not available. So clients can keep up working and the workflow requests will be processed as soon as the BPM server is back online.

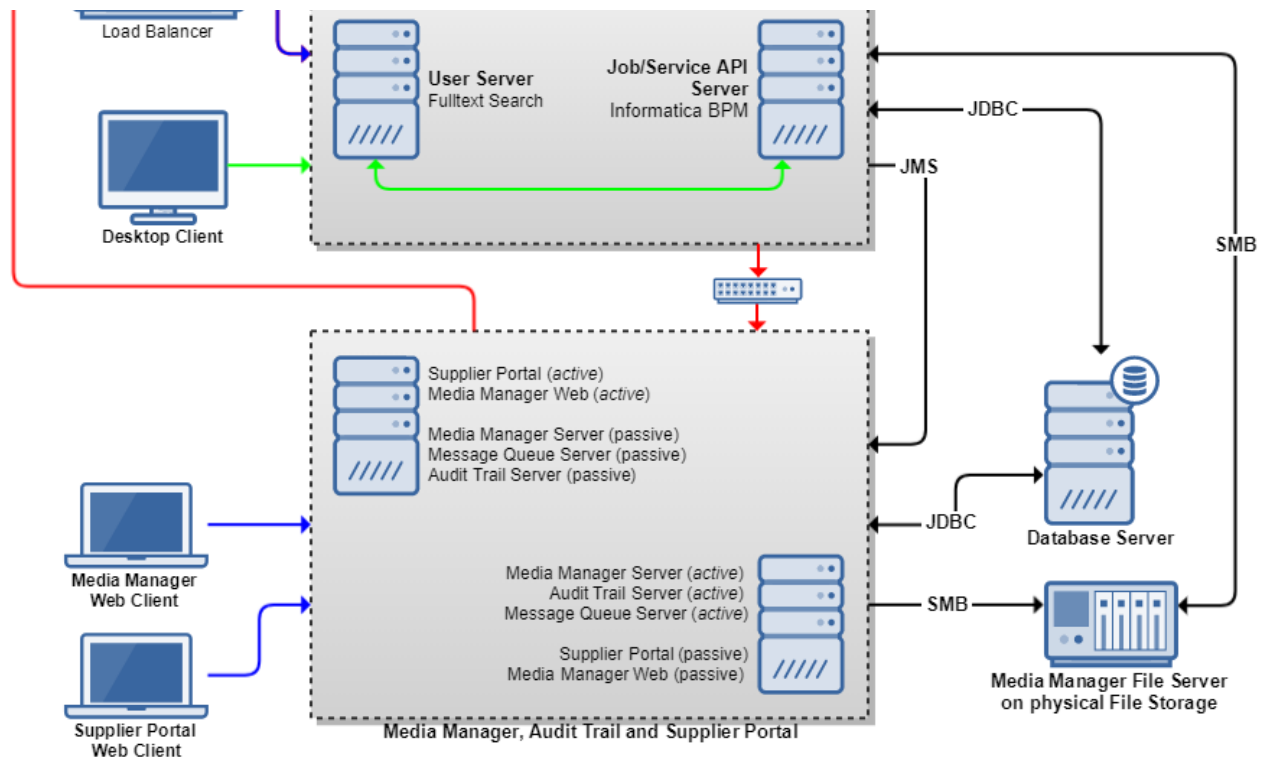
7.3.12 Message Queue Server

By default Informatica MDM - Product 360 uses the Apache Active MQ which can be also be deployed in a high availability mode, please refer to the available documentation for Apache ActiveMQ or any other 3rd party message queue system

7.4 High Availability without Load Balancing

As described earlier, the Informatica MDM - Product 360 deployment has a maximum of flexibility. The following is an example of how you could reach a highly available system without the need for laying out each module as a multi-server environment. Of course, the load scenarios for this kind of deployment are limited, but it might be sufficient for a large group of customers.





7.4.1 Application Server Cluster

Since the Informatica MDM - Product 360 Application server has a build-in fail-over logic (as described above), it's a good practice to just have two server instances. One for clients, one for jobs. If either of those crashes, the other one will take over his responsibility in the meantime. To be close to transparent for the clients, you would need to have a Load Balancer for this cluster - so all web client requests and service API requests are automatically routed to the server which is still working. The routing for the desktop clients is build in already. It's important that **all** HTTP traffic is routed through the Load Balancer otherwise the interconnection between the modules might not work.

7.4.2 Media Manager, Audit Trail and Supplier Portal

The services for Media Manager Web, Media Manager Server, Supplier Portal and and Audit Trail server are installed on two hosts. Each service is part of an OS cluster configuration in which some of the services are active on one host, and passive on the other. In case of a failure of either of the hosts, all services which are active on the failed one will be moved to the other host. So during a failure the services are still up and running. Depending on the available hardware they might be slower since the host needs to handle now all services, but they are still available. For the sizing of the hosts it must be considered that each host needs at least as much memory as all services together require in their minimum specification.

7.4.3 Media Manager File Server

The Media Manager File Server is a share of a Windows or Linux based server system. Please refer to an OS specific cluster solution. To prevent data loss use a professional backup solution or use at least a raid system for your shares.

7.4.4 Database Server

Please refer to the deployment manuals for the Oracle or MS-SQL Server databases for high availability and fail-over deployment strategies.

8 Limitations

8.1 Number of Servers in the PIM Server cluster

A PIM Server cluster can in theory hold an unlimited number of servers. In theory means that currently there are no build-in limitations on the number of servers - but the synchronization and communication overhead will increase with the number of servers. Please contact Informatica Support if you plan to use more than 4 application servers (not counting the servers of the other clusters like Media Manager or Audit Trail!).

8.2 Database Performance

As you can see in the diagrams, the database seems to be the next logical resource bottleneck. As of today this can only be solved with vertical scalability on the database. Please contact your DBA for details on how to achieve the needed performance on the database.



We do not guarantee that sizing parameters fit on virtual hardware. Our sizing recommendations are developed and measured on non virtual machines. Please note that performance on virtual environments can differ and is therefore not guaranteed. However, we still recommend to use current state of technology virtualization environments to be able to grow the servers also vertically. It's just important that the CPU cores from the sizing are really available in a physical way and not shared with other applications. A close monitoring on the host hardware is needed to inspect and adapt the needed resources to the everyday needs of the customer.

9 Sizing Calculator

The sizing calculator is provided as a easy to use excel document which provides a rough sizing estimation for a given customer scenario.

Download

Please always make sure to use the latest version of the calculator. Do not use a local copy from a previous project. Make sure you have the latest and greatest version since we constantly optimize the sizing calculation based on new information. Please contact the Informatica Support or your local Informatica representative so they send you the latest version.

The current Version of the Sizing Calculator is **4.1**.

9.1 How does it work?

The calculator uses various sources of information, like benchmark results for Product 360 and others. Together with answers of the questionnaire in the excel a few key sizing indicators are returned.



Please, in no case, treat the results as the absolute truth since they can only be estimations based on limited information and assumptions we did which are again based on benchmarks and customer experience. We recommend to check the results and discuss them with us before you send them directly to the customer. Questions like "on how many servers do we distribute the cores" are not to be answered with a simple excel file. They need to be discussed since every customer is different.

9.2 The Excel Sheets

9.2.1 Introduction

General introduction on the sizing calculation. Please read this carefully.

9.2.2 Questionnaire

Questions whose answers are needed in order to provide a meaningful sizing estimation.

Objects

Estimate the number of products, items as "Number of Product 360 objects"

The sum of all products, variants, and items. If additional custom entities are planned, please count them in.

Estimate the average number of attributes per object

The average number of attributes per object. Multiple qualifications of an attribute like "short description" are counted multiple times. E.g. short description for English and short description for German would be treated as two attributes.

Estimate the portion of dynamic attributes as part of the attributes above

Product 360 has the notion of "dynamic attributes" which can be defined by runtime. Since those attributes are treated differently and need to be stored differently we would need to know how many of the total number of attributes are those dynamic ones. Dynamic attributes are typically also mapped to structure group features. Examples would be "Color", or "Size". Please discuss this with the customer as this is a very important figure.

Number of Data Quality Rule Configurations per Object (average)

This must only be answered in case the Data Quality Engine is used. Please provide the average number of data quality rule configurations per item. Please note that this is not the number of raw DQ rules, but the number of DQ rule *configurations* within Product 360. A single DQ rule can be used multiple times! Here you specify how often a DQ rule is used (aka configured) within Product 360. Typical numbers range from a few dozen, up to over a hundred in case you're integrating with data pools like GDSN for example (the standard rule set for the GDSN accelerator has about 100 rule configurations). Note: This setting has a direct and significant impact on the needed memory of the application servers.

Users

Estimate the average number of parallel, "hard and daily working", users

In earlier versions of the sizing calculator we called those users "Full Time Equivalent". It basically means that those users spend their daily business work mainly working directly with PIM. It's not, checking now and then a product and modifying now and then a field, it's really, working one item by item, making mass operations like search and replace and multi selection classification or cumulative attribute editing. Typically those users use mainly the Desktop Client - but it might very well be that they use the web client. So it's not the client we're asking for here, it's the kind of user.

Estimate the average number of occasional users

An occasional user can be anything from accessing the system occasionally every few days to a few times a day just checking some data or approving some tasks. This is a big range and therefore should be thought about carefully. Typically occasional users don't have a Desktop Client installed, most times they only work with the web client. So as a rule of thumb, if you don't have a better value, take the number of web client licenses and subtract those which access the system only once a week or even less.

Estimate the number of objects modified by the users per day

This is a tricky one to catch. It's a general estimation on how much work all your users manually can do during a day. It's important for a estimation of audit trail file size and also for the job servers in combination with data quality.

Jobs

Estimate the number of objects imported in Supplier Catalogs per day

We differentiate between imported objects in supplier catalogs and the master catalog since usually a merge process is involved in the first one. As a rule of thumb, multiply the number of suppliers with an average catalog size. If you don't have any supplier catalogs, leave the field empty.

Estimate the number of objects directly imported in the Master Catalog per day

Please sum up all direct imports in the master catalog. This includes 3rd party applications using our import hot folder to integrate, as well as manual imports.

Estimate the number of objects modified from external applications (Service API) or customizings

Please sum up all external write access like 3rd party integration as well as all customizing jobs which process objects and modify them during this. Like some custom price calculation or attribute inheritance logic etc.

Estimate the number of objects exported per day

Number of exported objects means, the distinct sum of all assortments which are exported at a day, independent of any channel. So in case an item is exported into multiple channels, it count's just once here.

Estimate the number of channels to which the objects are exported

Define the number of export relevant channels the objects get exported.

Digital Assets

Estimate the average number of digital assets per Product360 object

This defines only the real, high quality assets, not the derivatives which are created out of those.

Estimate the average size of a digital asset in MB

The difference between a high quality picture with a few MB, or big a Photoshop file with multiple layers with 30 or more MB. This figure is very important for the estimation of the media asset file storage.

Plan the number and type of derivatives to be generated per original image

Define the average number of derivatives to be generated from the digital assets

Estimate the average size of a derivative asset in KB

Typically derivatives are sized in kilobytes. Pictures for the web shops are usually not bigger than 40 - 100 Kb.

Do you need the Media Manager Web Application?

The Media Manager Web Application provides a Web UI to managing Media Assets. If you need a separate web access for Media Assets only, then you have answer this question with Yes.

Modules

Audit Trail

Yes if you want to use the audit trail module. This makes it necessary to think about the database schema size and you need to have the extra audit trail application service.

Supplier Portal

Yes if you want to use the Supplier Portal module. Typically the supplier portal is deployed on a seprate machine as it's an own application facing the internet.

Web Search

Yes if you want to use the Web Search module which provide a google like search inside the Product 360 web client. Note that the search has extra hardware requirements.

Informatica Data Quality

Data Quality, or, DQ, is tightly integrated into the Product 360 application server. We assume that each modification of an object triggers also at least a single DQ check that particular object.

Informatica BPM

Yes if you want to use the Informatica BPM module. Informatica BPM is an own stand alone Informatica product. Please see the sizing documentation for Informatica BPM about any sizing details.

Printable Report

Number of Cores

In contrary to earlier sizing calculators we do not return a simple "use this machine with this amount of cores" result. Instead, we provide the needed number of cores and memory. It's up to the consultant or partner together with the customer to decide how those CPU cores are distributed across multiple servers, or even a single server.

Memory and Disk space

The memory and disc space needs are "per server". In case you decide to split the load on multiple servers, each of them needs the given amount of memory and disc space.

Database Server

As everything in the calculator, the number of cores in the database and the memory requirements are to be treated as a rough estimation. Especially here this is true. The sizing calculator can not replace the sizing recommendations of database vendors. We strongly recommend to make a separate sizing for the database based on the vendor's documentation. Please contact your DBA for help on this.

Database Storage

We recommend to use SSD drives for the database in order to have the best performance possible.