

07-02-02

60392999 06/28/02

APPROV

06/28/02
J1040 U.S. PTO

J1040 U.S. PTO
60/392999
06/28/02

PTO/SB/16 (8/96)

Approved for use through 01/31/98. OMB 0651-0037

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Attorney Docket No. 3399P103Z

Type a plus sign (+) inside this box [+]

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

INVENTOR(s)/APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE NAME/ INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
Andersson	Tomas	L.	Tävelsås, Sweden
Bergljung	Karl	Martin	Växjö, Sweden
Skagne	Anders	L.	Växjö, Sweden
Lund	Tomas	G.	Växjö, Sweden

TITLE OF THE INVENTION (280 characters max)

Product Catalog

CORRESPONDENCE ADDRESS (including country if not United States)

Jordan M. Becker

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

12400 Wilshire Boulevard, Seventh Floor

Los Angeles, California 90025-1030

Telephone: (408) 720-8300

FAX: (408) 720-8383

ENCLOSED APPLICATION PARTS (check all that apply)

☒ Specification	Number of Pages 27	☐ Small Entity Statement
☐ Drawing(s)	Number of Sheets	☒ Other (specify) Express Mail Certificate

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)

☒ A check or money order is enclosed to cover the filing fees

☒ The Commissioner is hereby authorized to charge
filing fees and credit Deposit Account No. 02-2666

Filing Fee Amount (\$) 160.00

This invention was made by an agency of the United States Government or under contract with an agency of the United States Government.

☒ No

Yes, the name of the U.S. Government Agency and the Government Contract Number are:

Respectfully submitted,

SIGNATURE

Jordan M. Becker

DATE

6/28/02

TYPED or PRINTED NAME: Jordan M. Becker

REGISTRATION NO. 39,602
(if appropriate)

Additional inventors are being named on separately numbered sheets attached hereto

12/01/97

- 1 -

PTO/SB/05 (12/97)

Netflix v. VideoLabs

IPR2023-00628

Netflix. Ex. 1008

EXPRESS MAIL CERTIFICATE OF MAILING

EL821850631US

"Express Mail" mailing label number: _____

Date of Deposit: June 28, 2002

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Julie Arango
(Typed or printed name of person mailing paper or fee)

Julie Arango
(Signature of person mailing paper or fee)

6/28/02
(Date signed)

Serial/Patent No.: _____
Client: Openwave Systems Inc.
Title: Product Catalog

Filing/Issue Date: _____

BSTZ File No.: 3399P103Z
Date Mailed: 6-28-02

Atty/Secty Initials: JMB/jaa
Docket Due Date: ***

The following has been received in the U.S. Patent & Trademark Office on the date stamped hereon:

- ☐ Amendment/Response (____ pgs.)
- ☐ Appeal Brief (____ pgs.) (in triplicate)
- ☐ Application - Utility (____ pgs., with cover and abstract)
- ☐ Application - Rule 1.53(b) Continuation (____ pgs.)
- ☐ Application - Rule 1.53(b) Divisional (____ pgs.)
- ☐ Application - Rule 1.53(b) CIP (____ pgs.)
- ☐ Application - Rule 1.53(d) CPA Transmittal (____ pgs.)
- ☐ Application - Design (____ pgs.)
- ☐ Application - PCT (____ pgs.)
- ☒ Application - Provisional (27 pgs.)
- ☐ Assignment and Cover Sheet
- ☒ Certificate of Mailing
- ☐ Declaration & POA (____ pgs.)
- ☐ Disclosure Docs & Orig & Copy of Inventor's Signed Letter (____ pgs.)
- ☐ Drawings: _____ # of sheets includes _____ figures
- ☐ Other: _____

- ☒ Express Mail No.: EL821850631US
- ☐ _____ Month(s) Extension of Time
- ☐ Information Disclosure Statement & PTO-1449 (____ pgs.)
- ☐ Issue Fee Transmittal
- ☐ Notice of Appeal
- ☐ Petition for Extension of Time
- ☐ Petition for _____
- ☒ Postcard
- ☐ Power of Attorney (____ pgs.)
- ☐ Preliminary Amendment (____ pgs.)
- ☐ Reply Brief (____ pgs.)
- ☐ Response to Notice of Missing Parts
- ☐ Small Entity Declaration for Indep. Inventor/Small Business
- ☒ Transmittal Letter, in duplicate
- ☐ Fee Transmittal, in duplicate

Check No. 51056
Amt: \$160.00
Check No. _____
Amt: _____

PROVISIONAL UNITED STATES PATENT APPLICATION

Product Catalog

INVENTORS:

Tomas L. Andersson
Karl Martin Bergljung
Anders L. Skagne
Tomas G. Lund

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(408) 720-8300

Docket No. 3399P103Z

"Express Mail" mailing label number: EL821850631US

Date of Deposit: June 28, 2002

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Julie Arango
(Typed or printed name of person mailing paper or fee)

Julie Arango
(Signature of person mailing paper or fee)

6/28/02
(Date signed)

Openwave Systems Inc.

Inventor Name: Tomas Andersson

Group Name:

IDEA PROPOSAL

Problem

Organize the product catalog in a way it make sense when it comes to provisioning without enabling the complex view of such storage model to subscribers.

To make sense, the catalog should have the following characteristics:

- Meta-data (ie information regarding a product) is related to a product, not its physical implementation.
- A product may point to one or more implementations
- Each implementation is developed for a target device, or a specific device capability set.
- It is possible to pick products, based on available implementations and how well they match the target device capabilities.

Known approaches

Content management is traditionally focused on the content itself and thereby its physical characteristics. There are similar implementations in place at certain wireless operators, but they offer a product-binary scheme, rather than a product-implementation-binary scheme.

Another aspect of content management systems, is that content is stored with the meta-data itself and the meta-data is related directly to the physical content. In our scheme, the meta-data is related to the product (which is a proxy for a suitable physical content).

Comments

As the ODM is targeting provisioning in a wireless world with plenty of devices accessing the system constantly, the product catalog's storage model is built to meet the requirements of filtering content depending on the device capabilities. It has also been built to guarantee the quality of the delivered content. A third issue implemented by the product catalog is the content location transparency, the subscribers should not be aware of whether the content is located internally or remote.

The product catalog is the repository for the content within the iMAP, whether it physically resides in the iMAP or it's remotely located, the product catalog manages the "pointer" to the content, in other words the product catalog is abstracting the actual address to any content provided. The product catalog should be used by different roles so the view of the catalog is different depending on which role the current client has.

A product consist of one or more implementations which in turns consists of one or many binaries, see figure 1. Separating a product into several implementations enables the provider to deploy different implementations collected under the same name, i.e. a chess game implemented as both a MIDLET as well as an APPLET. Each of those implementations targets different devices. If the user connects with a device which supports one or both of the implementations the product should be discovered, else it won't be shown in the users product menu. This is accomplished by using the matching mechanism in the DiscoveryManager. The matching is based on the least common denominator approach where suppliers specifies only those attributes and values that should be supported unless the model or vendor of a device. A supplier should focus on significant attribute answering questions as, "Does the device supports sound output?", "Has it a color screen?", "Can it handle MIDLETS?" etc and compile those minimum requirements to one of the implementations. However the suppliers aren't restricted to be generic they can still specialize the requiriements to be very restrictive, like stating it only runs on a Ericsson T68. Using this approach ensures that the users is provided with the most suitable content at the moment, but still they just see the product as the other mechanism is hidden by the system. Of course this can also cause problems while a subscriber purchase a product from one device which matches the requirements of at least one implementation but later try to download it from another device currently not supported. The system will not deliver any content to that device while it cannot assure that delivered content is suitable.

The product catalog provides carrier to enable and disable parts of the content within the same product, still having that product online. This is achievable by a filtering mechanism that disallow discovering of unapproved content, ensuring both the quality of content towards the subscribers as well as the accessibility to the product towards the suppliers.

Ellipsus Systems

infiniteMAP **Feature Specification** **Product Version**

Version 1.0

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

Revision History

Date	Version	Description	Author
	0.1	Draft	Tomas Andersson
	1.0	Ratified	Martin Bergljung

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.3.1	Definitions	4
1.3.2	Acronyms	4
1.3.3	Abbreviations	4
1.4	References	5
1.5	Overview	5
2.	System Description	6
2.1	Overview of the InfiniteMAP system and the Discovery Manager	6
2.1.1	Overview of the Discovery Manager GUI	6
2.2	Use-Case view for the Discovery Manager	6
3.	Features	7
3.1	Requirements	7
3.2	Feature Set: Discovering a Product Set	8
	Feature	8
4.	Architecturally Significant Design Packages	9

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

Detailed Design Specification

1. Introduction

1.1 Purpose

This document describes the detailed features for the Version, which is called a *major feature set* (MFS). It uses the requirements from the functional requirements specifications as *feature sets* (FS) and details them into features.

1.2 Scope

The Version finds the products which can be retrieved from the product catalog. The DM also has to maintain an index/cache to boost performance when accessed.

1.3 Definitions, Acronyms and Abbreviations

The definitions, acronyms and abbreviations from [A.1] apply.

1.3.1 Definitions

Term	Description
Filtering	The process in which a set of products are filtered and all or a subset of these products are selected for further processing.
Bundle	A product description which may contain information from both product, implementation and binaries
Bundle Repository	A client view of the product catalog
Discovery	Finding products based on a set of criterias
Context	The current set of criteria

1.3.2 Acronyms

Acronym	Description

1.3.3 Abbreviations

Abbreviation	Description
DM	Discoverer Manager
PC	Product Catalog
JSR	Java Specification Request
GUI	Graphical User Interface

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

1.4 References

Id	Description (title, version, date, publisher, source)

1.5 Overview

This FS is a detailed design document below [R.1], which is the overall functional requirement specification of the MAP.

Each feature is assigned a priority (pri). This priority can take the following values:

Value	Priority	Meaning
L	Low	This feature can be deferred to later in the project or even to a later release.
M	Medium	This feature can be deferred to later in the project but should be implemented prior to first release.
H	High	This feature should be implemented as soon as possible to verify that it can be realized, or to be able to implement other aspects of the system, which rely on this requirement.
N/A	Not Applicable	This feature is not subject to prioritization since it is of general character and affects the overall development project.

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

2. System Description

2.1 Overview of the InfiniteMAP system and the Version

The Version implements the Discovery process, by taking all arguments required and determines what products match the context and should be discovered (returned). The DM accesses the product catalog to get a set of products, but it keeps it's own internal index of those products, which implies it shouldn't have to access the PC constantly, but only when it wants to reindex or when a product is added, updated or deleted. The Version should implement the interface `javax.provisioning.BundleRepository` to be compliant with the JSR-124 draft(s).

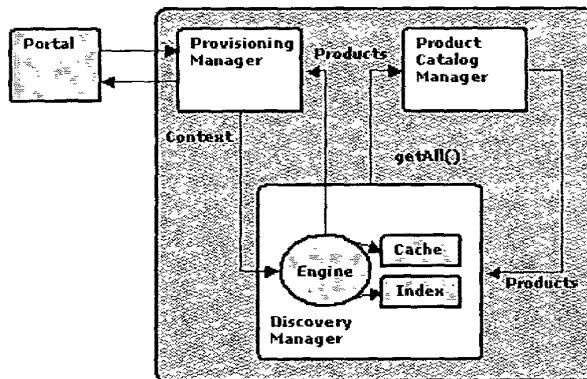


Figure 1, Version overview

2.1.1 Overview of the Version GUI

N/A

2.2 Use-Case view for the Version

N/A

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

3. Features

3.1 Requirements

The following requirements have been extracted from [R.1] and form the *feature sets* for this module. One requirement can result in several Feature Sets. Several requirements can also fit in one single feature set.

Req #	Requirement	Pri	Description	Notes	Owner	Feature Set
	Filter products		It shall be possible to filter products before sending them to the portal.			
	Filter plug-ins		It shall be possible to plug-in filters which implements different filtering mechanisms.			

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

3.2 Feature Set: Discovering a Product Set

The Version should be able to discover products based on filter settings.

#	Feature	Notes	Pri	Owner
1	Find the products matching a set of search criteria		H	
2	Find the product with an id		H	

infiniteMAP	Version: 1.0
Feature Specification	Date:
MAP-002-FS-23 – Product Version	

4. Architecturally Significant Design Packages

The following packages should be used when implementing the Version:

```
com.ellipsus.imap.product.discovery  
com.ellipsus.imap.product.discovery.vo  
com.ellipsus.imap.product.discovery.dao  
com.ellipsus.imap.product.discovery.ejb
```

Ellipsus Systems

infiniteMAP Feature Specification Product Catalog Manager

Version 1.2

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

Revision History

Date	Version	Description	Author
	0.1	Draft	Tomas Andersson
	1.0	Minor formatting changes	Tomas Lund
	1.0	Added delegation to Content Analysis manager. Added deployed flag to be set by provider. Added product bundles	Tomas Andersson
	1.01	Minor formatting changes	Tomas Andersson
	1.02	Revised the feature list	Tomas Andersson
	1.02	Ratified	Martin Bergljung
	1.1	Added support for localized strings. Removed any requirements which involved the pricing and configuration of pricing models, while it has been transferred to LicenseManager	Tomas Andersson
	1.1.1	Added feature set for: - Internationalization	Martin Bergljung
	1.1.2	Removed features from Internationalization feature set not belonging in the pc Changed the version numbers	Tomas Andersson
	1.2	Removed requirements of CAM execution from within the PC Described the DEPLOYED and APPROVED states Brief discussion of event generation	Tomas Andersson

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.3.1	Definitions	4
1.3.2	Acronyms	4
1.3.3	Abbreviations	4
1.4	References	5
1.5	Overview	5
2.	System Description	6
2.1	Overview of the product catalog	6
2.1.1	Overview of the product catalog manager	7
2.1.2	Overview over content location strategies and benefits	7
2.1.3	Content status	9
2.1.4	Generation of event	9
2.2	Overview of the product catalog GUI	9
3.	Features	9
3.1	Requirements	9
3.2	Feature Set: Managing a product	13
3.3	Feature Set: Approving content	13
3.4	Feature Set: Bundling products	13
3.5	Feature Set: Managing deployed products	13
	Feature	13
3.6	Feature Set: Discovering products	14
3.7	Feature Set: Downloading content	14
3.8	Feature Set: Localizing a product	14
	Feature	14
4.	Architecturally Significant Design Packages	15
4.1	com.ellipsus.imap.product.catalog	15

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

Detailed Design Specification

1. Introduction

1.1 Purpose

This document describes the detailed features for the Product Catalog Manager, which is called a *major feature set* (MFS). It uses the requirements from the functional requirements specifications as *feature sets* (FS) and details them into features.

1.2 Scope

1.3 Definitions, Acronyms and Abbreviations

The definitions, acronyms and abbreviations from [A.1] apply.

1.3.1 Definitions

Term	Description
Product	A product consists of a product description and the implementations of the product.
Implementation	The implementation of a product consists of two parts – the description and its binaries. The implementation description is typically downloaded to the client and drives the downloading of the binaries of the implementation.
Binary	A binary is the content delivered to the client that constitutes the product.
Content	Content is a synonym for description of the implementation and all binaries that belong to an implementation.
Catalog	Product catalog short
Bundle	A collection of products bundled together by some business rule. This definition conflicts with the definition of a Bundle in the JSR124 so it will change in the future

1.3.2 Acronyms

Acronym	Description
MFS	Major Feature Set
FS	Feature Set
GUI	Graphical User Interface
CRUD	Create, read, update and delete operations on entities

1.3.3 Abbreviations

Abbreviation	Description

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

1.4 References

Id	Description (title, version, date, publisher, source)
[R.1]	Functional Requirement Specification, MAP-002-RS-01, Ellipsus Systems.

1.5 Overview

This SD is a detailed design document below [R.1], which is the overall functional requirement specification of the MAP.

Each feature is assigned a priority (pri). This priority can take the following values:

Value	Priority	Meaning
L	Low	This feature can be deferred to later in the project or even to a later release.
M	Medium	This feature can be deferred to later in the project but should be implemented prior to first release.
H	High	This feature should be implemented as soon as possible to verify that it can be realized, or to be able to implement other aspects of the system, which rely on this requirement.
N/A	Not Applicable	This feature is not subject to prioritization since it is of general character and affects the overall development project.

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

2. System Description

2.1 Overview of the product catalog

The product catalog is the repository for the content within the iMAP, whether it physically resides in the iMAP or it's remotely located, the product catalog manages the "pointer" to the content, in other words the product catalog is abstracting the actual address to any content provided. The product catalog should be used by different roles so the view of the catalog is different depending on which role the current client has.

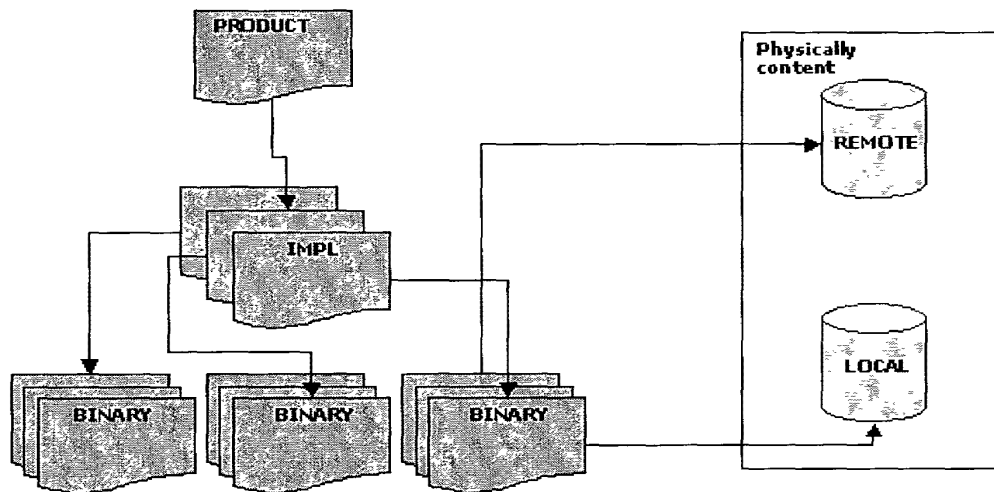


Figure 1, Product hierarchy

A product consist of one or more implementations which in turns consists of one or many binaries, see figure 1. Separating a product into several implementations enables the provider to deploy different implementations collected under the same name, i.e. a chess game implemented as both a MIDLET as well as an APPLETT. Each of those implementations targets different devices. If the user connects with a device which supports one or both of the implementations the product should be discovered, else it won't be shown in the users product menu. This is accomplished by using the matching mechanism in the device capability manager. Each product belongs to a category, which are defined by the owner of the system, categories are defined in a tree structure, where you can select any node to tie the product to.

Another feature provided by the product catalog is the bundling of existing products. These bundles can be offered to customers in the same way as products, i.e. you bundled a collection of songs from an artist and call the bundle ArtistX – Best of. The bundling is performed by someone at the server side, like an admin or similar. Depending on the view of the catalog, a product or one of it's subcomponent can have different status. The provider can determine whether a product should be deployed or not. Once deployed the admin have to approve the content by calling the Content Analysis Manager and use the result from the call to set the approved status. When the product is deployed and it's content is approved it's available for the consumers. Descriptions and names of products is stored as localized strings, which the supplier defines. This implies that there could be several descriptions and names for the same product, which a client could select among when it comes to presentation.

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

2.1.1 Overview of the product catalog manager

The product catalog manager responsibility is to manage the catalog, which means expose methods such as the users can change, read, update and delete products and product subcomponents. This manager also allows and disallows content to be viewed in the system, until they are approved by the admin. Approval can be either manual or automatic based on a contract between the provider and the owner. Unapproved products can be viewed using the product catalog manager, but they won't be discovered by users.

Separating administration into different roles make sense while a provider should only be able to view it's own products, not a competitors. Customer care will have limited access to the catalog as they are suppose to have less skill than a administrator and therefore should have a more controlled access to destructive operations. Separation between roles is enabled thru different interfaces which are determined before accessing the business layer, typically in a servlet's `service()` method.

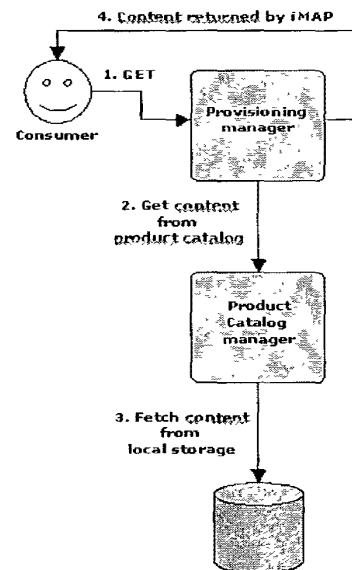
As a user (thru provisioning manager) performs a discovery or download process the product catalog manager's responsibility is to provide a view of the existing product catalog to the users/consumers based on a specific context, including user preferences like preferred language, category subscription among others. The service methods should be design for fast execution while the amount of users are high and the user experience is important to make the users to return to the provisioning system, in other words **performance is prioritized when dealing with discovery and download operations**. The product catalog manager exposes methods to the provisioning manager, or any other manager or entity like a filter engine or a indexed cached of the product catalog, which enables discovering of products, downloading descriptor files and downloading binaries.

2.1.2 Overview over content location strategies and benefits

As iMAP promote remote location of binaries it reveals three strategies of storing content in iMAP. These three different is called local, semi-remote and remote.

2.1.2.1 Local storage

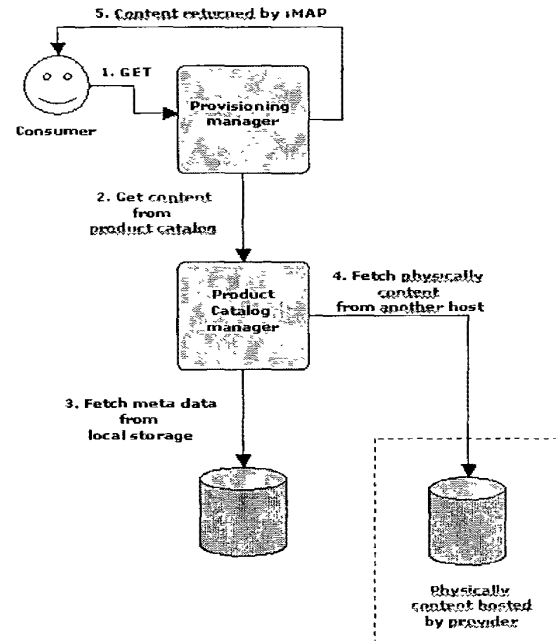
Providers have to upload all content that should be used and this content is stored at the local storage, typically a database which supports LOB's. This approach is less complex from the iMAP perspective and once the content is uploaded the iMAP got full control of it. Storing the content in iMAP increase performance while fetching content but in a scalability perspective this is a less efficient strategy where users always access the Provisioning Manager whatever they want to do. Another issue is the quality of the content which can be guaranteed by iMAP.



InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

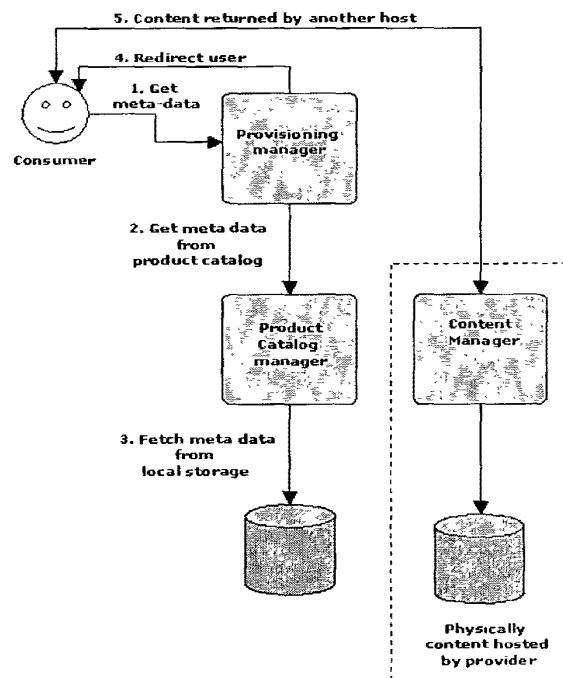
2.1.2.2 Semi-remote storage

When using semi-remote storage the providers can keep their physical content at another host, ensuring the legal aspect of handling content. As iMAP wants to ensure the quality of the downloaded content, it has to get the content from the providers host and ensure it fulfills the requirements for delivery to the end-user. This approach isn't efficient while it includes two requests for content, one to iMAP and one to provider's host.



2.1.2.3 Remote storage

To achieve better scalability combined with the providers aspect of having control of the content, iMAP offers totally remote storage for content which means the user want get the content from iMAP, but instead it will get the content directly from the provider. This approach frees the iMAP when it comes to download, but on the other hand iMAP can't guarantee the quality of service, while it has no control weather the content is online and not malicious.



InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

2.1.3 Content status

The content in the catalog may be in a couple of states where each state enables or disables functionality for different users.

2.1.3.1 Approved

Implementations are either approved or unapproved. When created it's always unapproved until it's specifically set to approved. An unapproved implementation won't be enabled for download meaning that it will be filtered out when the discovery process is executed.

2.1.3.2 Deployed

A product can be either deployed or undeployed. When deployed the product is available for discovery, which implies that the content is enabled to all subscribers using the vending machine. When a product is deployed the VM has to ensure that the content is approved according to the setup configurations of the VM. Once a product is deployed all changes of content of that product will be analyzed by the CAM. A deployed product can contain zero or more approved implementations.

2.1.4 Generation of event

To inform other components in the system of changes made on products or any of its subcomponents the PCM generates events for certain operations. For example when a product is deployed the PCM should generate a ProductDeployedEvent to trigger a automatic analysis of the newly deployed product. This analysis probably approves some of the implementations which in turn generates an ApprovedImplEvent from the PCM. Those events are only for descriptive use and could be renamed or even not exist in the implementation.

2.2 Overview of the product catalog GUI

As the catalog is separated to support different roles its GUI is also separated.

3. Features

3.1 Requirements

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

The following requirements have been extracted from [R.1] and form the *feature sets* for this module. One requirement can result in several Feature Sets. Several requirements can also fit in one single feature set.

Req #	Requirement	Pri	Description	Notes	Owner	Feature Set
1	General	H	The product content format shall be transparent to the MAP and treated as arbitrary byte content.			3.2
2	Topics	H	The user shall be able to select topics, sub-topics and even specific products within topics.	Note – the product catalog is the best place to store and maintain such topics (categories)		3.2
3	Language	L	The user shall be able to select preferred language(s), which shall be supported by downloadable products. <i>Note-</i> . This requirement will affect the user view. It is an addition to the general requirements in.	Note – implies that provider must be able to select supported language(s) for content		3.2
4	History log	H	The MAP shall have a history log, where events caused by user interaction are stored. <i>Note-</i> . The purpose of this log is to facilitate customer care.	Note – implies that the product catalog has to generate logs for new products, updated logs, etc		3.2
5	Device independency	H	The virtual storage points to product name only and shall be transparent to the actual implementation of the products. <i>Note-</i> . This makes it possible to acquire a product, pay once and then download its various implementations to several different device types.	Note – the products got to have a human readable name		3.2 3.5
6	Device type	H	It shall be possible to tailor the application menu based on device type.	Note – the product catalog must store information about the supported capabilities for a certain product		3.2 3.5
7	Best fit	H	The menu shall be based on a “best fit” where the products are matched against execution capabilities. <i>Note-</i> . There might be two version of an application,	Note – implies that a product can have one or more different implementations targeting different device types		3.2 3.8

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

			one provisions a URL, the other an applet. If the device is capable of running an applet as well as a URL, the applet is the preferred product to download and consequently also to present in the menu.			
8	Move product	L	It shall be possible to move products between topics.	Note – the provider must be able to select among predefined topics to tie to the product		
9	General updates	L	It shall be possible to perform product updates that apply to many records in secondary storage, such as products, user profiles, events a s o.			3.2 3.5
10	Configuration model	H	The MAP shall use a deployment model where applications are either actively downloaded to the MAP via the deployment interface, or downloaded via a batch process. <i>Note-</i> . This affects the requirements on the API.			3.2 3.6
11	Version control	L	The MAP shall, when possible, query each product provider for new versions or new products periodically as a batch process. <i>Note-</i> . Could run once per day for example.			
12	Approval	H	All submitted products shall be approved prior to making it available to the consumers.			3.3
13	Approval by contract	H	It shall be possible to set up a contract between a product provider and the MAP, so that approvals are not necessary.	Note – the contract is set up in the user management but is consider by the product catalog manager when products are deployed or updated		3.3
14	Notification	L	When a new product has been submitted, the administrator shall be notified.			3.2
15	Product accessibility control	H	It shall be possible to disable malicious products from being displayed to the consumers, if it is detected that it is faulty and/or holds malicious content.			3.3
16	Provider usage	M	The product provider shall be able to interact with the MAP using the API to automate the deployment process.			3.6

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

			<i>Note-</i> . This could mean that the MAP can query product providers for new/revised content. This requirement is supporting the requirements for product providers.		
17	Product view management	H	The MAP shall have a product view management interface for the administrator use.		3.4 3.5 3.8
18	Product management	H	The MAP shall have a product management interface for the administrator/customer care use.		3.5

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

3.2 Feature Set: Managing a product

This feature set is targeting providers in iMAP who should be able to deploy products manually via the web

#	Feature	Notes	Pri	Owner
1	Edit the <i>attributes</i> for a product		H	
2	Edit the <i>attributes</i> for a implementation		H	
3	Edit the <i>attributes</i> for a binary		H	
4	Change the <i>deployed status</i> for a product		M	
5	Generate the <i>new product event</i> for a product deployment		L	
6	Generate the <i>update product event</i> for a product update		L	
7	Generate the <i>removed product event</i> for a product removal		L	

3.3 Feature Set: Approving content

Product catalog offers the ability to approve content by the analysis result by the CAM and the status of the provider., ie trusted/untrusted. If the content analysis is successful a trusted providers content is automatically approved. If the analysis fails, the product admin can set approve status manually based on the result. An untrusted providers content will be marked as “analyzed” and the product admin can approve the content based on the analysis result.

#	Feature	Notes	Pri	Owner
1	Get the <i>analyzed content</i> for a product		L	
2	Set the <i>approved status</i> for a trusted providers successful analyzed product.		L	

3.4 Feature Set: Bundling products

Product catalog offers the ability to bundle products into purchasable bundles of products. The catalog is responsible for administer and providing bundles to each component which may ask for bundles.

#	Feature	Notes	Pri	Owner
1	Edit the <i>attributes</i> for a bundle of products		L	

3.5 Feature Set: Managing deployed products

The manager shall implement an administrative interface to enable administration of products and their subcomponents. This interface must support common operations like create, read, update and delete (CRUD). The interface methods is targeting the administrator and customer care but could be used by the provider as well.

#	Feature	Notes	Pri	Owner
1	Set the <i>approve status</i> for content		H	
2	Disable the <i>content</i> for a product with malicious content		L	

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

3.6 Feature Set: Discovering products

Product catalog is responsible for deliver and search among products during discovery. The actually flow is controlled by another module, but supported by the following features available in the catalog

#	Feature	Notes	Pri	Owner
1	Get the <i>product set</i> for the product catalog		H	

3.7 Feature Set: Downloading content

Product catalog is responsible for deliver content to users during the download process. The actually flow is controlled by another module, but supported by the following features available in the catalog. The catalog is holder of the descriptor file as well as the binaries.

#	Feature	Notes	Pri	Owner
1	Get the <i>descriptor file</i> for selected implementation		H	
2	Get the <i>binary content</i> for a selected binary		H	

3.8 Feature Set: Localizing a product

The product catalog have several fields that need to be localized such as the name of the product, the categories that the product is connected to etc.

#	Feature	Notes	Pri	Owner
1	Localize the <i>name</i> of the product		H	
2	Localize the <i>description</i> of the product		H	
5	Localize the <i>image</i> of the product		H	

InfiniteMAP	Version: 1.2
Feature Specification	Date:
MAP-002-FS-13 – Product Catalog Manager	

4. Architecturally Significant Design Packages

4.1 `com.ellipsus.imap.product.catalog`

Main package which should contain interfaces to the Product Catalog manager