

Product Indexer

EHP Dev Team

Version History

Date	Version	Description
2010/04/13	1	Created
2010/07/14	2	Added requirements section, proposed architecture
2010/08/04	3	Updated requirements, use cases, proposed architecture
2010/09/09	4	Added architecture diagram

Table of Contents

Problem	4
Scope	4
Stakeholders	4
Background	5
<i>Events</i>	<i>5</i>
<i>Addons</i>	<i>5</i>
<i>Products</i>	<i>5</i>
Requirements	6
<i>General</i>	<i>6</i>
<i>Association</i>	<i>6</i>
<i>Authoritativeness</i>	<i>7</i>
<i>Notification</i>	<i>7</i>
<i>Search</i>	<i>7</i>
Use Cases	8
<i>Notify User</i>	<i>8</i>
<i>Search Catalog</i>	<i>8</i>
Proposed Architecture	9
<i>Product Catalog</i>	<i>9</i>
<i>Product Indexer</i>	<i>9</i>
<i>Association</i>	<i>9</i>
<i>Product Summary</i>	<i>9</i>
<i>Product Modules</i>	<i>9</i>
<i>Diagram</i>	<i>10</i>

Problem

Each product currently distributed through the EHP web site uses separate delivery, indexing, and notification methods. The lack of a standardizes product distribution process is a barrier to integration for new products and a maintenance nightmare for existing products.

Scope

Product Indexer receives products from Product Distribution. This includes maintaining a product catalog and notifying users when its catalog is updated.

Stakeholders

Product Producers - Any person or application that creates products, often automated processes. Both internal USGS and external non-USGS producers exist, particularly regional networks.

Product Consumers - Any person or application that uses products, often automated processes. The public is often considered the consumer, but products are also used internally.

Existing QDM Users - A subset of Producers and Consumers who rely on QDM. This group is identified because the Indexer replaces QDM and must provide equivalent functionality.

Background

This document assumes familiarity with Product Distribution design document.

Events

Each network assigns a unique identifier (ID) to each event. They reference this ID when updating event information, or when linking other information to that event. When events from the same network have different IDs, they are separate events.

When multiple networks locate the same event, each network assigns their own unique ID to their message. Because of this, systems must compare messages from separate networks to see if they refer to the same event.

When messages from more than one network refer to the same event, one is chosen as the preferred solution. QDM uses ANSS authoritative regions to choose which message is preferred.

Addons

Addons associate products to events using event IDs. This generally works, but has several limitations.

QDM doesn't maintain a catalog of addons. External systems must manage an addon catalog separately, including versioning and choosing rules for addons for non-preferred events.

Producers may link a product to an event ID that is not preferred. External systems must track non-preferred IDs to properly associate this information and process updates.

Producers may send a product without also submitting an event or addon. Because associations occur at the event level, these products do not correctly associate.

Products

RecentEqs uses QDM as its source of event information, so most product authority is based on event authority. However, products are often related to a non-preferred event and are not automatically associated to the preferred event. An example of this is tsunami warnings being associated with solutions that are never preferred.

ShakeMap defines separate authoritative regions from Hypocenters. The ShakeMap indexer was not designed to handle authoritative regions. Special rules had to be implemented within ShakeMap to prevent this duplication.

Requirements

General

- Must be platform independent
QDM is platform independent.
- Should not require external dependencies
The indexer should be self contained and easy to install for most users.
- Should run on reasonable commodity hardware
Users should not require special hardware to run this application.
- Must maintain a catalog of events and products
- Must receive products from Product Distribution/EIDS
EQXML/QuakeML messages contain products.
- Must support any type of product
New types of products that haven't been created yet should still be associated when properly tagged.
- Must be able to catalog a subset of products
Some users are only interested in a catalog of ShakeMaps.
- Must provide flexible archiving policies
Some users want past 7 days, some want all products for all time.
- Must integrate with external systems for Notification and Search

Association

How events and products are identified and related

- Must support association using location in space and time
QDM association rules are default.
- Should support association using direct links
Product generators can explicitly link one product to another.

Authoritativeness

How related products are ordered

- Must support product specific authoritativeness rules
Location is important in determining origin authoritativeness, while other factors may be more important when determining magnitude authoritativeness.
Additionally, ShakeMap uses different authoritative regions than Hypocenters.
- Must use ANSS authoritative rules for Hypocenters by default
For products that do not define product specific authoritativeness rules.

Notification

How external systems are notified when the catalog is changed.

- Must provide notification when catalog is updated
Reduce overall latency by eliminating polling delays.
- Notifications must be filterable
Users may only be interested in new events or certain types of products

Search

How external systems query the catalog

- Must allow searches for products and events
- Must allow searches based on product properties

Use Cases

Notify User

Users are explicitly notified when changes are made to the catalog, and do not need to perform comparisons to understand how the catalog has changed.

Examples of this use case:

- PAGER creates a product when a preferred ShakeMap is added to the catalog.
- ENS sends email notifications when events are added or updated in the catalog.
- Web pages are created after new products are added to the catalog.

Search Catalog

Users search the catalog to find events and products that are interesting.

Examples of this use case:

- A regional network operator searches the catalog to find events and products submitted by their regional network.
- A monitoring application searches the catalog to find when each network last submitted a message.
- An automated system retrieves a list of all preferred events for the past month.

Proposed Architecture

Product Catalog

The product catalog differs from the QDM catalog in that it handles all types of products, not only origin and magnitude products. When discussing the product catalog, an event is a group of products that share a similar location in space and time. Events form when the first product arrives for that location, often an origin or magnitude product.

Product Indexer

The product indexer receives products from Product Distribution and EIDS, and maintains a product catalog. The indexer compares events before and after products are added to the catalog to determine how the catalog has changed, and notifies listeners in order to react to these changes. A listener is provided that uses the Product Distribution Command Line API for notification.

Association

The indexer builds on QDM associations. Event parameters (location and time) are still used when available. The indexer also supports direct association using product links.

To associate a new product, the indexer finds related products in the catalog. When products are found, there is already an event. When products are not found, an event is created.

Product Summary

The indexer uses product summaries to perform associations. Summary information is derived from PDL and EIDS message headers. Summaries include an identifier, event parameters, links to other products, authoritative weight, and product specific properties; essentially a product without its contents. This information is used to find related products, build events, and when searching the catalog.

Event parameters typically come from origin and magnitude products. However, other products, like ShakeMap, include these parameters in their product content. These parameters allow products to associate even when separate ids are used.

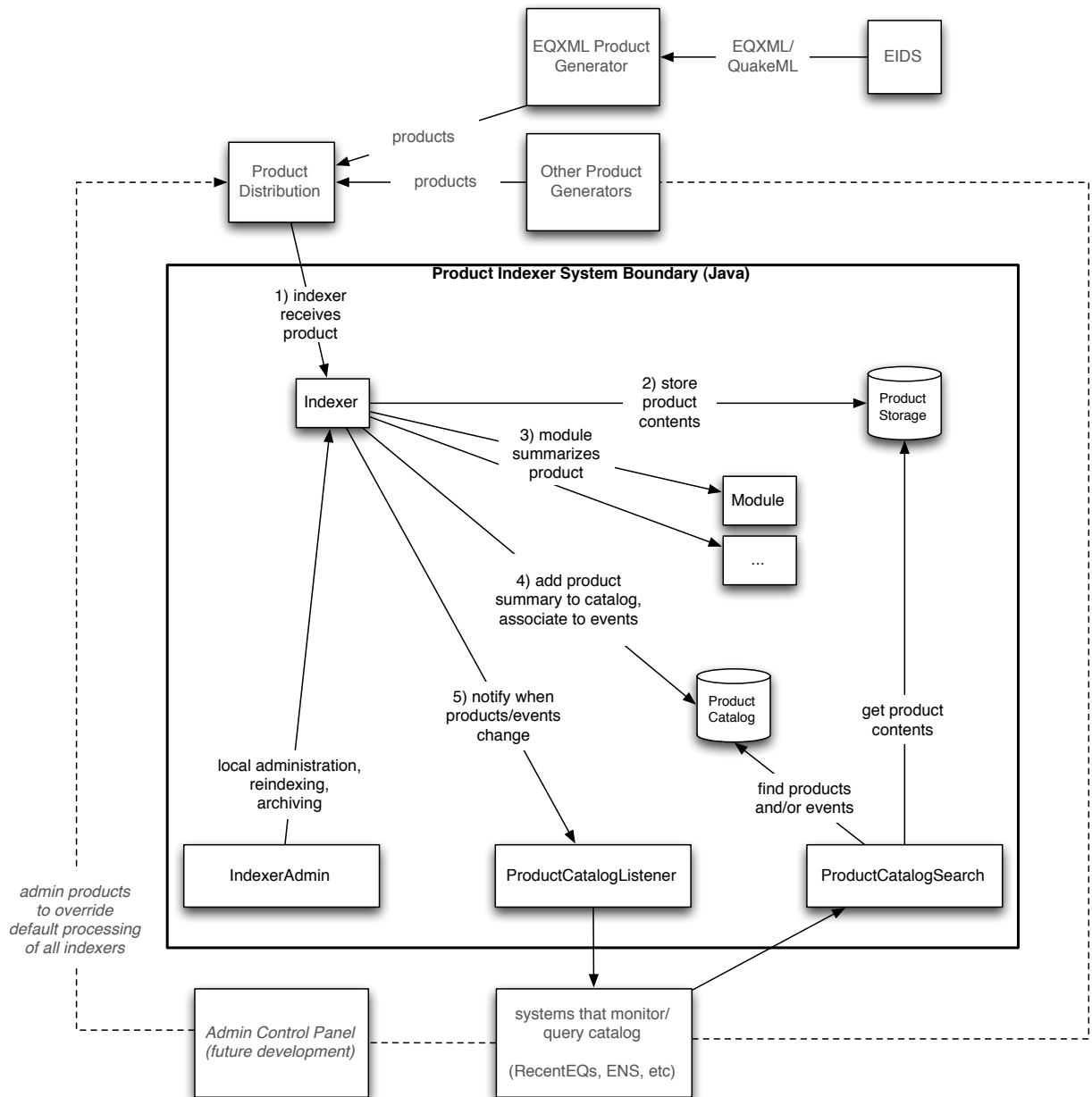
Authoritative weight is a number. When two products are considered duplicates, the product with higher authoritative weight is chosen.

Product Modules

The indexer allows each type of product to create a custom module to provide product specific functionality. Modules can customize product summaries to override default authoritativeness rules, and add properties used when searching.

A default module is provided that is used when a custom module is not available. Product producers who encode properties when sending products usually do not need a custom product module.

Diagram



Starting at the top are product generators sending products using product distribution. The numbered steps (1-5) show how the indexer receives products from product distribution, updates its catalog, and notifies external systems. Additionally, local search and administrative interfaces are provided to access information in the catalog.

An additional, planned system is the Admin Control Panel, which can be used to create ad hoc products and override associations made by default in all indexers.