

The SUSE Linux Feature Tracking Tool

Cornelius Schumacher
Klaas Freitag



Novell.



SUSE Feature Management

- Motivation
- Architecture
- Clients
- Demonstration
- Conclusion

Motivation



Motivation

- Managing Features for SUSE Linux Products
- Feature request come in many flavours
 - Feature requests by partners
 - Requirements by product management
 - Enhancement suggestions by users
 - ...
- Several thousand requests
- Development for multiple products in parallel



Involved Persons

- Partners
- Technical Account Managers
- Product Managers
- Project Managers
- Team Leaders
- Developers
- Documentation Writers
- Quality Assurance



Process

- Request filed by technical account manager on behalf of partner, product manager, engineers
- Review by product and project managers
- Review by team leader
- Implementation
- Test
- Release



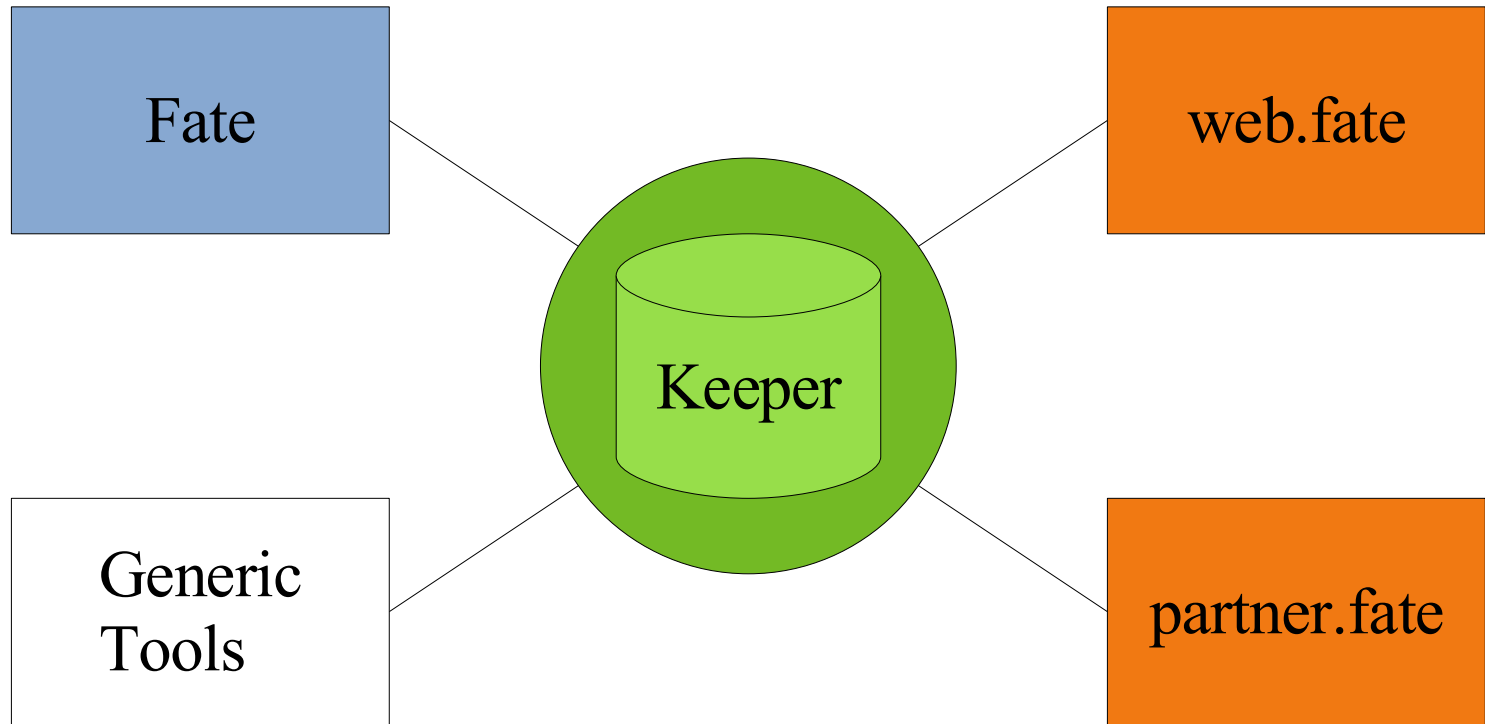
Feature Data

- Id
- Title
- Categories
- Description
- Product context (product information, status, priority)
- Actors (involved persons)
- Partner context (partner information, business case)
- References
- Impact on documentation and QA
- Comments, discussion
- ...

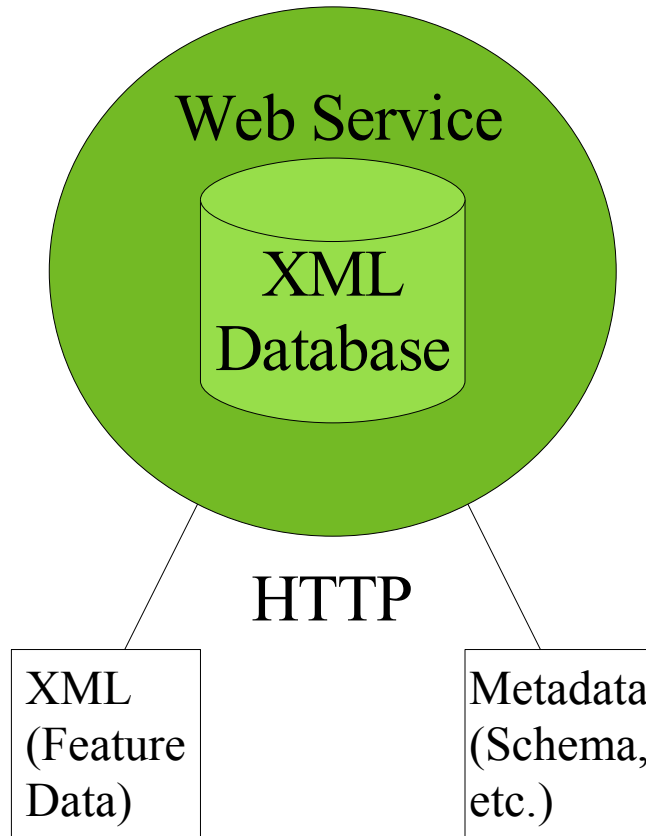
Architecture



SUSE Feature Management System



Keeper



- Data Storage
 - XML Database (Berkeley DB XML)
- Client Interface
 - Web Service (REST, XML over HTTP)
 - Data Access (XML)
 - Schema Access (XML Schema)
 - Metadata Access
- Email change notification
- Administration Interface



Client Access Protocol

- Protocol
 - REST-based web service
 - XML over HTTP
 - XQuery interface
 - Optimistic locking (conflict detection on write)
- Data
 - Feature data (XML)
 - XML Schemas
 - Style Sheets (XSLT) for HTML and text views
 - Metadata (categories, product lists, ...)



Additional Client Access Features

- Versioning
 - Change history
 - Access to previous revisions of documents
- Diffs between revisions
 - XSLT converts XML to text
 - text diff
- Different data representations
 - HTTP Accept headers
 - XML, HTML, Text
- Schema update handling
 - XSLTs for transforming data on schema upgrade



Feature Data Example (1/3)

```
<feature schemarevision="4" id="300666" revision="1" >
  <category>Kernel</category>
  <category>SUSE Linux Enterprise Server</category>
  <title>Exciting Feature</title>
  <description>
    <richtext>
      <p>This is the most exciting feature of the next
      release.</p>
    </richtext>
  </description>
  ...
```



Feature Data Example (2/3)

...

<productcontext>

<product><productid>SLES-10</productid></product>

<status><new/></status>

<priority>

<mandatory/>

<owner><role><productmanager/></role></owner>

</priority>

</productcontext>

...



Feature Data Example (3/3)

...

```
<discussion>
```

```
<comment created="2006-06-23T01:36:26" id="1" >
```

```
<author>
```

```
<person><email>cschum@suse.de</email></person>
```

```
</author>
```

```
<richtext>
```

```
<p>Very important, because customers really want it.</p>
```

```
</richtext>
```

```
</comment>
```

```
</discussion>
```

```
</feature>
```



Client Access API

- URL: `http://keeper.suse.de/feature/123`
- HTTP Request
 - Header
 - `GET feature/123`
 - `Accept: text/xml`
 - ...
 - XML Body
 - `<feature>...</feature>`
- HTTP Response with XML Body
 - `<status>...</status>`



Basic Client Access API

- Read all features
 - GET feature
- Read features for query
 - GET `feature?query=/feature[category=Kernel]`
- Read one feature by id
 - GET `feature/123`
- Update one feature
 - PUT `feature/123`
- Create new feature
 - POST feature



Extended Client Access API

- Metadata

- GET `schema/feature`
- GET `resource/feature.xslt`

- Versioning

- GET `feature/123/history`
- GET `feature/123/2`
- GET `feature/123/diff?oldrev=2&newrev=3`



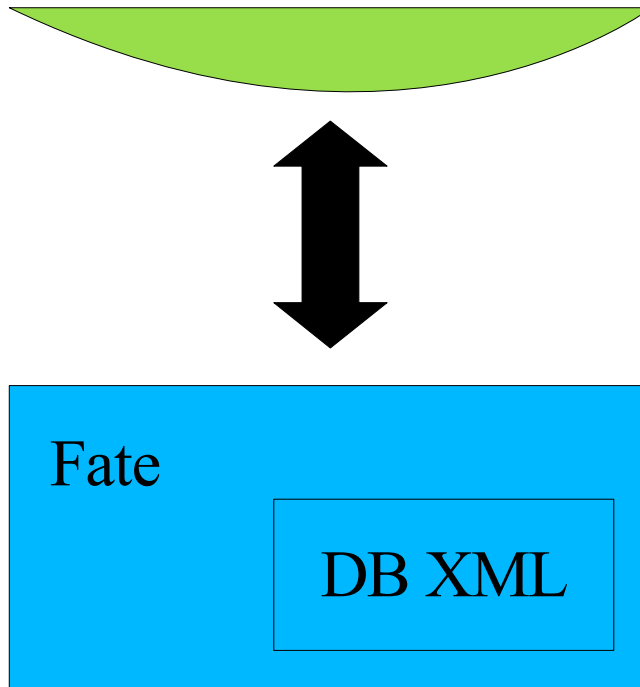
XQuery

- Standard query language for XML data
- Keeper implements full XQuery spec
- Clients mostly use only subset querying feature objects
- Examples:
 - Features with a given category
`feature[category="Kernel"]`
 - All new features
`feature[productcontext/status/new]`
 - All features with actor "abc" and no priority
`feature[actor/person/email="abc" and not productcontext/product/priority]`

Clients



Fate



- Rich native GUI client
- Queries
- Browsing
- Feature View
- Feature Editor
- Validation
- Offline Access (work in progress)
- Printing



Fate Implementation

- Uses the KDE framework
 - kioslaves for HTTP access
 - KHTML to render HTML
- libxslt for XSLT transformations (XML to HTML or text)
- Local caching of metadata
- Local database for offline access (work in progress)
- Generic view without schema dependencies
- Editor tailored to feature schema
- Validation for edited features (“business logic”)
- Printing via external tool (based on Docbook and FOP)



Fate Screenshot

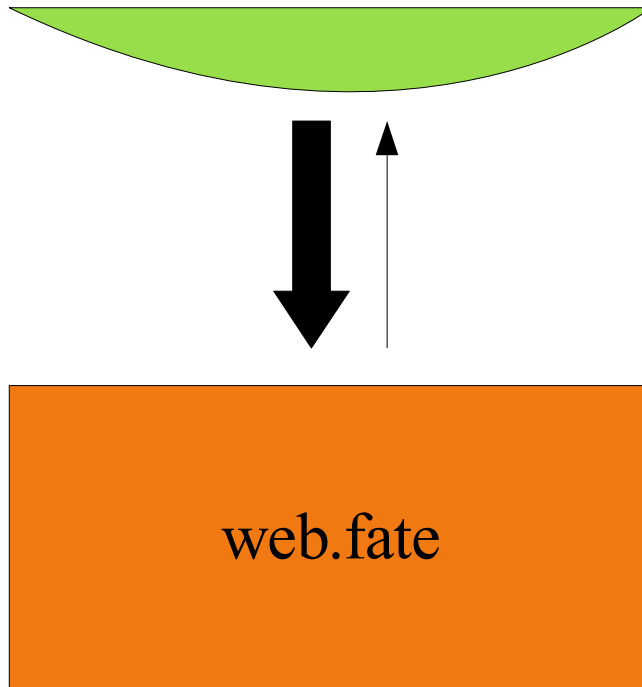
The screenshot shows the Fate web interface in a browser window. The window title is "cschum@suse.de as Interested - Fate". The interface is divided into several sections:

- All Features:** Includes tabs for "Flat" and "Tree", and two filter sections. The first filter shows "444 of 444 Categories" with a search box. The second filter shows "3280 of 3280 Features" with a search box.
- Category List:** A list of categories including NFS performance improvements, NFS server, NFSv4 Support, NIS Server/Client Issues, Novell Client components, Novell eDirectory authentication support, Novell GroupWise connectivity, Novell Linux Enterprise SDK, and Novell Open Enterprise Server.
- Feature List:** A table with columns "Id" and "Title". The first row is highlighted: "300666 Exciting Feature". Other rows include "300667 Support for terabit ethernet", "300668 Kernel 3.0", "300669 Wireless Power", "300670 Focus-follows-mind", "300671 4D Desktop", "300672 Suspend to Internet", "300673 Coffee machine module for YaST", and "300674 Performance Optimizations".
- Feature Detail View:** Shows details for feature #300666: "Exciting Feature".
 - SLES-11:** Status: New; Priority: Mandatory for Requester.
 - Persons:** Requester: cschum@suse.de
 - Categories:**
 - [SUSE Linux Enterprise Server](#)
 - Description:** "This is the most exciting feature of the next release."
 - Discussion:**
 - #1: [cschum@suse.de](#) () [[reply](#)]: "How long will it take to implement?"
 - #2: [cschum@suse.de](#) () [[reply](#)]: "Should be ready for christmas."
 - [\[add comment\]](#)
 - Footer: "entered 2006-06-23 01:35:54 by cschum", "revision 2, last modified 2006-06-23 09:17:40 by cschum"
 - Active Renderer: Pretty View (dropdown)
 - Buttons: [Save](#), [Vview](#), [Editor](#)

Reloaded feature #300674 (0.575 s)



web.fate



- Web Client
- Queries
- Browse
- Adding comments

- Java Server Faces
- Shares style sheets, etc. with Fate



web.fate Screenshot

The screenshot shows the web.fate application running in the Konqueror browser. The browser's address bar shows the URL `http://keeper.suse.de:8080/webfate/main.jsf`. The application header includes the text "web.fate" and a search box labeled "Features by id" with a "go" button. Below the header, there are tabs for "Navigation", "Filters", and "Queries (experts)". A sidebar on the left lists 3272 features, categorized into various groups such as Products (23), Linux Kernel (876), Base System (184), Installation (169), Update (22), New Software and Patch management (36), Configuration (244), Administration (29), Applications (194), Server Applications and Services (40), Software Development and Tools (33), Desktop (253), Multimedia (67), Mobility (41), High Availability (19), High Performance Computing (6), Security (57), Live CD (9), OEM Preload (13), Documentation (17), Internationalization (118N / L10N... (11), Design (8), Standards (9), Benchmarks (7), YaST Internal Changes (178), Language Bindings (3), User Interface (30), and Textmode specific topics (7). The "User Interface" category is expanded, showing sub-items like "Asynchronous events in NCurses", "Redesign of PAD widget", "Richtext widget bugs", "Selection sorting using package m...", "Nicer ncurses table widget", and "UTF-8 related bugs".

The main content area displays a list of features, with the first feature selected: "#120361: Richtext widget bugs". The feature details include:

- more info >>** Found 1 features, displaying page 1 of 1 / features 1 through 1.
- #120361: Richtext widget bugs**
- future:**
- Status:** New
- Persons:**
- Requester:** jm@suse.de
- Categories:**
 - Textmode specific topics
 - User Interface
 - YaST Internal Changes
- Description:**

Some improvements, maybe a redesign is required
See <http://bugzilla.suse.de/24312> , <http://bugzilla.suse.de/38026> , <http://bugzilla.suse.de/46410>
- Partner:**
- NDA:**

no NDA
- Discussion:**

no comments yet
[\[add comment\]](#)

At the bottom right of the feature details, there is a timestamp: "entered 2005-02-11 08:39:32 by inttools@suse.de revision 3, last modified 2006-04-21 18:56:06 by tschmidt".



partner.fate



Full Schema



Restricted Schema



- Web client for limited access
- Restricted access to subset of data
- Limited view
- Limited edit rights
- Perl implementation



partner.fate Screenshot

The screenshot shows a web browser window titled "SUSE Feature Tool - Konqueror <2>". The address bar contains the URL: `http://zwackel@boltzmann.suse.de/ft/index.cgi?rm=feature_show&id=300666`. The page content is as follows:

- Header:** SUSE partner.fate logo on the left, and "#300666: Exciting Feature" on the right.
- SLES-11:**
 - Status: New
 - Priority: Mandatory for Requester
- Persons:**
 - Requester: cschum@suse.de
- Description:**

This is the most exciting feature of the next release.
- Partner:**

SUSE
- NDA:**

no NDA
- Discussion:**
 - #1: cschum@suse.de () [[reply](#)]
How long will it take to implement?
 - #2: cschum@suse.de () [[reply](#)]
Should be ready for christmas.

[\[add comment\]](#)

entered 2006-06-23 01:35:54
revision 3, last modified 2006-06-23 14:17:57

SUSE Featuretool © 2004 SUSE LINUX Products GmbH · featureadmin@suse.de



Generic Clients

- Standard protocol allows to use standard clients
- Web browser to directly browse keeper
- HTTP aware network-transparent tools
 - kate for direct editing of raw XML data
- HTTP tools
 - curl
 - wget
- Writing clients is easy (guess why we already have three)

Demonstration

Conclusion



Conclusion

- Feature data schema
- Light-weight feature management process
- Client-server feature management system
- Generic HTTP and XML based server
- Several clients for different purposes

- Generalization to other schemas possible

- Open Source
 - Releases: Keeper: sxkeeper 1.0, Fate: fate 1.2
 - developer.novell.com/wiki/index.php/Sxkeeper
 - Mailing list: sxkeeper-devel@forge.novell.com

Novell®