ORE Goals

Juan J. García de Soria

Id

1 Introduction

We're observing, in our experience at work in big companies whose main business is providing and selling services to their customers, that a particular function of their operation, which is the rating and the billing of their customers' services, is being trusted to products and systems developed by third parties.

Typically these third parties are specialized on providing solutions for these specific needs, and sell prepackaged products (which are sold as flexible, general solutions) and a varying amount of consulting and integration services needed for the adaptation of their products to the environment and business processes in which they are expected to perform their functions.

Often, these companies deciding to implement these products can't afford or are not confident enough to either develop internally or contract externally an specifically architectured solution.

These companies are a small, niche market for a small number of vendors to place their products, that are typically fully proprietary, and with per-subscriber licence fees.

2 Motivation

It's our experience that these so called general solutions present a number of problems to the companies implementing them:

- Both the market and the number of vendors is small, so that the products that are provided are expensive and lacking quality due to the low competition level.
- The provided solution is not usually as general as needed, so that specific customizations are often required, thus incurring in development expenses anyway.
- The concepts these products were born around (for instance, telephone service rating and billing) usually get forced into different business models (for instance, broadband Internet access), thus forcing the implementation environment to adapt its processes to the product, in the cases that the product can't be adapted to the processes.

- These products usually are built on proprietary, non-standard, technologies, thus forcing these 'alien' technologies into the implementor's IT ecosystem and making further development or adaptation of the systems difficult. - The core, proprietary software, remains under the vendor only control, so that for adding major features to it, or even for fixing core bugs, the implementor has to depend fully on a single vendor, whose road-map may not match the implementor's.

Usually, every one of the problems described above is solved when a 'functionality market' is commoditized by one or more Free/Libre and Open Source Software (FLOSS) implementations. This has been the case with web and database servers, as well as operating systems. The Apache Web Server, jointly with the long list of high quality Open Source projects fostered by the Apache Foundation, are a fine example of inexpensive, universal, solid and adaptable software.

The main problem with the market for rating and billing solutions is that it's a small market, with a relatively little number of companies and organizations in the world having needs for an enterprise-level system. That makes for a small number of solution providers too, with little competition, so that the commercially developed solutions are not good enough.

No FLOSS effort has been realized so far in this direction. We think that's because FLOSS developers prefer to solve either personal, social, or technical problems. The rating and billing functions needed by a big services company are, though presenting some technical challenges, more of a financial or business problem.

As for some of us the implementation of the existing proprietary products has become somewhat of a personal problem :), we think it's time for a FLOSS alternative to be born, so that a community can be built around the problems in order to solve them.

3 Scope and functions

The intended scope for ORE includes the following functions:

- Service and Service Catalogs: ORE shall allow the operator to define the different services that the operator provides to their customers, managing the parameters needed in order to differentiate them. ORE shall also allow the operator to define rules about how, to whom, when and in which combinations may these services be sold and provided.
- Tariffs and Prices: ORE shall allow the operator to define what will be the price for service subscriptions and service usage for any of the services or combinations sold. These definitions would include the ability to specify formulas or procedures, when needed, in order to determine the cost of services. Discounts and special promotions shall be allowed too. Tax definitions shall be supported.

- Customer and User Accounts: ORE shall allow the operator to manage any needed information about its customers, including personal, financial and subscription information. This information shall be made easily available to external systems and modules (legacy systems, on-line interfaces, etc.), so that integration with those is as easy as possible.

- Provision of data to Service Elements: ORE shall be able to generate requests as customers subscribe new services or as their associated information changes, if needed, in order to inform the systems that provide the actual service about any needed change. For instance, when a customer subscribes an e-mail account service, the mail server that will host the customer's account would need to be notified, so that it allocates the disk space needed for the new account and activates it.

- Rating of usage records: ORE shall be able to rate the records by which the system measures the usage of the services that the customers do. That is, ORE shall be able to process those usage records and to calculate how much money (or how many points, or how much of any other kind of resource) to charge the user because of the usage of the service. ORE would use the rules defined in Tariffs and Prices in order to determine how much to charge. For instance, a phone call that is two minutes long, when applied a tariff of 0,10 EUR fixed plus 0,05 EUR per minute, would be rated to a total cost of 0,20 EUR, that will be added to the customer's balance. Rating in ORE shall allow also for the rating of subscription events (i.e. subscription or monthly fees), the progressive rating of progressing events (i.e. usage that is rated on the basis of small parts or units of usage as they are notified to the system), and the preauthorization of service usage prior to the actual service happening when operating in prepaid modes. Calculation and handling of taxes shall be performed by this step too.

- Storage of usage records: ORE shall be able to store and query rated usage records in a database, so that they may be queried or used when billing the customers. The definition and enforcement of data retention policies shall be allowed by ORE. For instance, an operator may be required by law to keep phone call records up to two years after the calls were done.

- Billing of customers: ORE shall be able to generate Invoices for each customer Account, so that they can be fed to any external system actually used to charge the customers. The information calculated here would consist of the aggregation, sorting and itemization of the rated usage records that a customer is to be charged for when a billing period ends. This information will be made available in a machine-readable format, as well as (optionally) in human-readable format for printing, mailing or on-line viewing.

4 Architectural and technological choices

Whenever making design decisions, the following criteria will be taken into account:

- Open technologies and standards, especially when they ensure interoperability and portability, will be used whenever possible. Taking this into account, Java has already been chosen as the main programming language for the project, because it's a mature and universal foundation for programming business applications, and because its portability allows for a wide range of hardware platforms to be used without significant porting costs. A number of Java-related technologies may as well be used. Other standards or standard practices like the usage of web application interfaces or the usage of XML for data interchange shall be considered widely.

- Existing Free/Libre and Open Source Software components shall be used when possible instead of developing specific software components for the project. FLOSS components exist for most of the infrastructure needs that ORE will present. Specifically ORE shall work on a FLOSS database manager (for instance, PostgreSQL), so that implementing ORE won't require a proprietary database (nor any other proprietary component). However, any interaction with the database component will try to be portable enough as to allow for the substitution of any reasonably standards-compliant commercial database, if that's preferred by a given implementor of ORE.

- Any function in the system that may reasonably require customization shall be made extensible. A proper balance must be the target here, so that the system is extensible, but usable with basic functionality out of the box. That would require us to implement a plug-in system for each one of the functional areas in ORE, while providing basic plug-ins with ORE so that basic functionality is supported without any need for further programming.

5 Licensing

ORE will be made available under the LGPL.