



# Smoothing the On-Ramp to Commercial

## Making it Easy for Open Source Users to Buy

*Larry Augustin, Managing Director, Augustin Ventures*  
larry.augustin@gmail.com

### **Director**

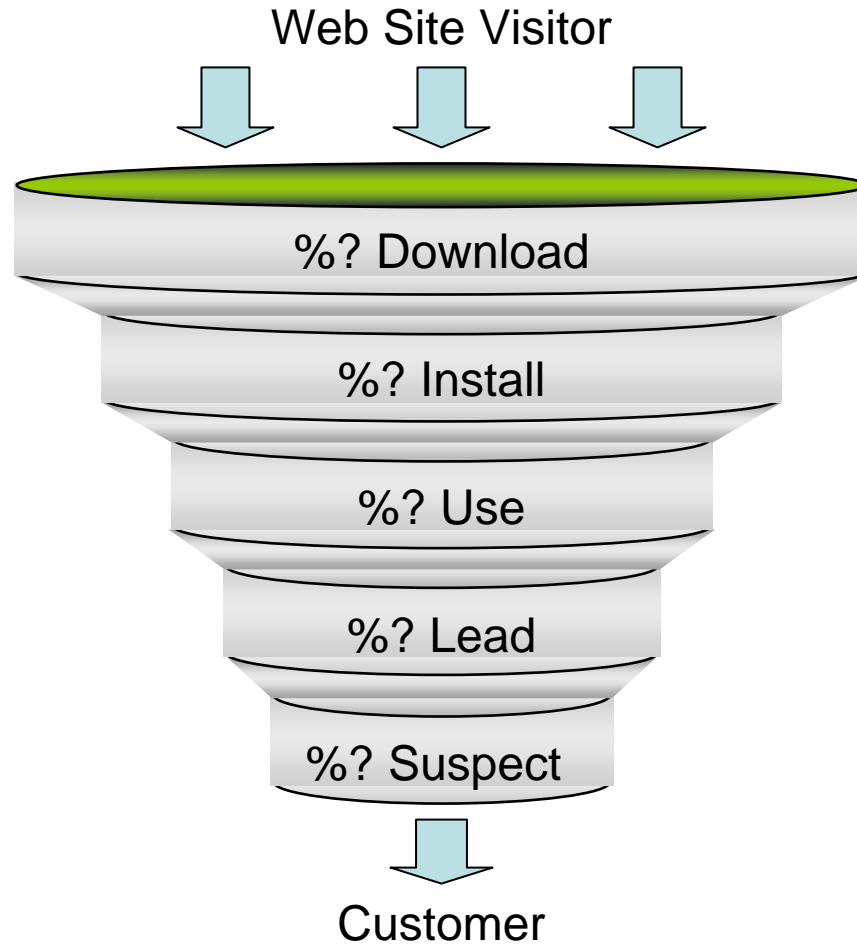
Compiere, DeviceVM, Fonality, Hyperic, Medsphere, Pentaho, SugarCRM

### **Investor/Advisor**

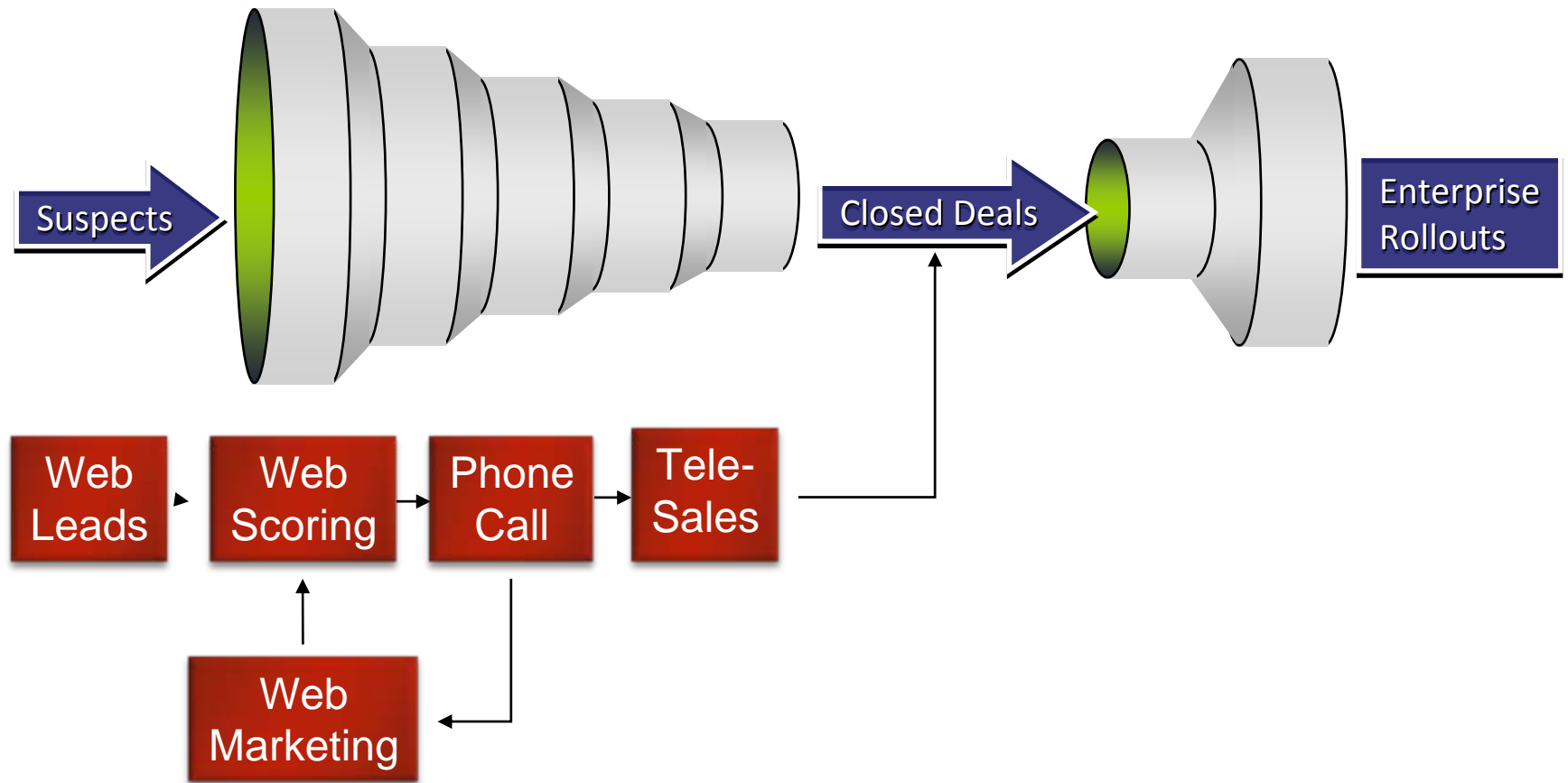
Appcelerator, Eloqua, GuideWire, ITeating, Mindtouch, MuleSource, Ohloh, REvolution, SpringSource, Vyatta, WSO2, Zend

**March 2008**

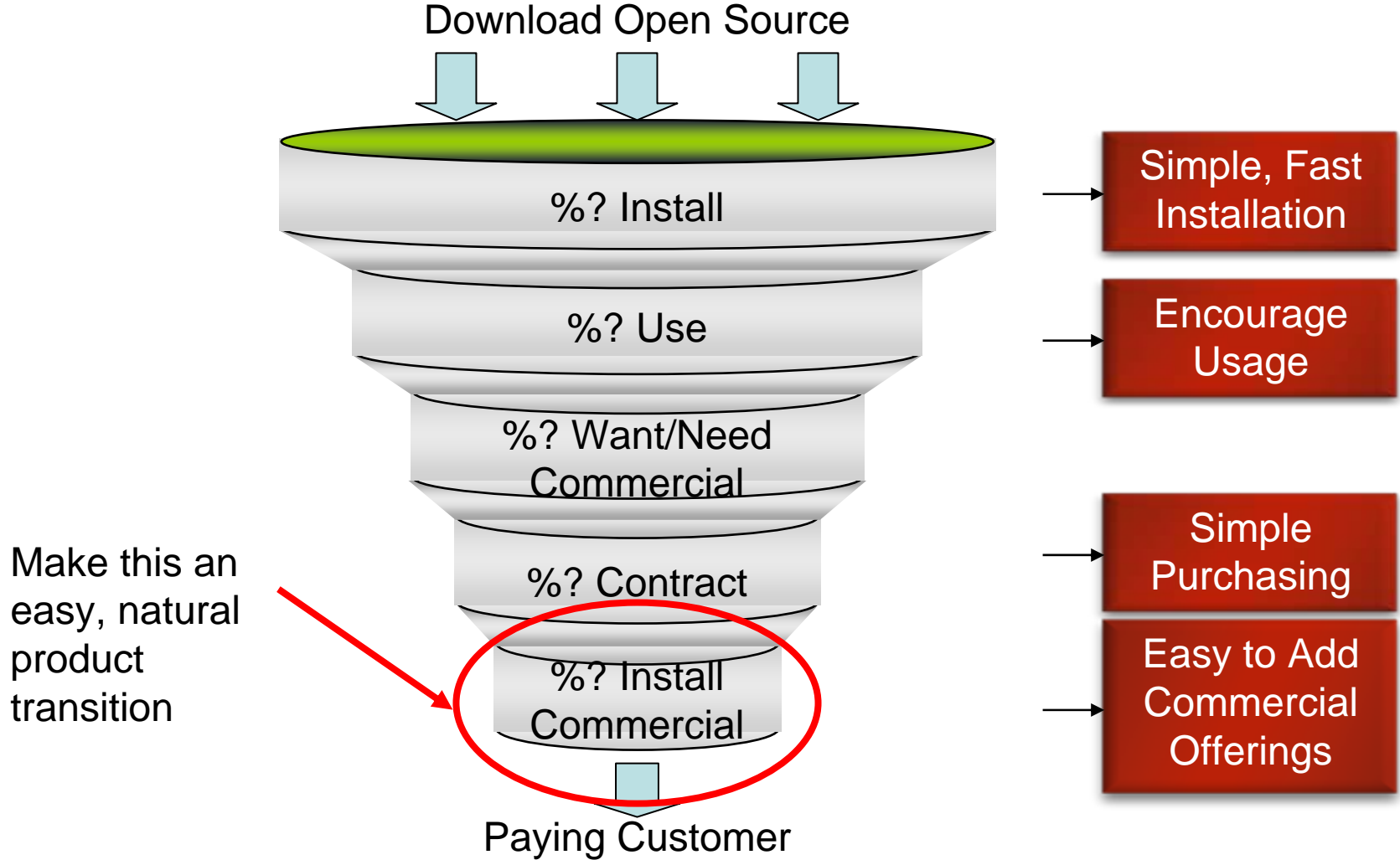
# Sales Funnel



# Closing Suspects



# Deployment Funnel



# Simplify Installation

# Fast, Simple Install

## Concept

- One-click immediate install
- Think in terms of minutes to usage, not hours
- One-click install for extensions from within the product
- Zero-config

## Examples

- MySQL 15-minute rule
- Bundle installers
  - Bitrock (SugarCRM, Pentaho, many users)
- Upgrade with product UI
  - SugarCRM, Zimbra, DNN
- Hyperic: Zero config sys mgmt

## Pros

- Instant gratification

## Cons

- Eliminates ability to sell ease of acquisition and ease of install as a value add

# Virtual Appliances

## Concept

- VMWare, Xen virtual appliance
- RPath virtual appliance

## Examples

- Many
  - Medsphere, SugarCRM

## Pros

- Consistent, integrated stack
- Very simple for the customer to manage
- Works well for larger, complex installations with lots of dependencies

## Cons

- Large downloads
- Works well for demos, but limited traction in real deployments

# Get Embedded in Distros

## Concept

- Bundled in Ubuntu, Debian, Red Hat, SuSE, etc.
- Install via the standard package mechanism
  - Apt-get, Aptitude, Yum,

## Examples

- MySQL
- Zend (PHP)

## Pros

- Instant gratification
- Leverages existing packaging and distribution mechanisms

## Cons

- Embedding in distributions loses visibility

# Navigating Purchasing

# Fly Under the Radar

## Concept

- Sell multiple small deals
- Deals can be expensed or put on credit cards
- Terms don't need legal review
- Approach large companies at the workgroup level
  - Organization is a set of divisions

## Examples

- Anyone who has a sub \$5K entry level product
- SugarCRM entry price at \$119/user/year

## Pros

- Contract size not material
  - Terms don't need legal review
- Fast sales cycle
  - “Corporate Impulse Buy”
- Reduced dependence on large deals
  - Smoother revenue

## Cons

- Need to sell many deals to make large numbers
- Need to have an efficient, automated back-end

# Plan for Product Lifecycle

## Concept

- Contract is not just one product purchase
- Structure the contract to include other/future product offerings as well
- Include a pricing schedule that reflects, allows and encourages greater adoption
- Plan for full lifecycle of adoption

## Examples

- Anyone that includes a pricing schedule via an appendix in their contract
- Anyone that includes a product list via an addendum or appendix
- Ability to add SOWs

## Pros

- One legal review
- One purchasing review

## Cons

- May lock in unfavorable pricing
- May lock in unfavorable deal terms

# Fixed Terms Format

## Concept

- Cover letter specifying product, price and other major terms.
- Back-end license presented “under glass” i.e. not negotiable
- Tends to work with smaller deal sizes and limited/no legal review

## Examples

## Pros

- Eliminates legal negotiation in the process.

## Cons

- May surprise some customers later in the process

# Easy Commercial Upgrades

# Same Binary

## Concept

- Open Source binary is the same as the commercial product binary
- Add value in the form of management, commercial add-on modules, support & services, etc.
- Opposite of the “certified binary” model
- Proprietary vendors: “license key upgrade”

## Examples

- JBoss
  - Same binary downloaded from SF.net by developers used in production
- Counter example: Fedora/RHEL
  - Fedora installation needs a complete migration to new RHEL OS
- MSFT: one binary, different license keys (“feature scaling”)
- MySQL, Pentaho

## Pros

- No reinstall, re-test, etc.
  - No migration
- Consistent messaging
  - Open Source product is production ready

## Cons

- Eliminates one way to add value for pure Open Source products
  - If open source binary is production ready, why do I need the commercial version?

# Modularize

## Concept

- Enable extensions, add-ons via a modular API

## Examples

- DotNetNuke
  - Large add-on community
- Hyperic
  - Agent Plug-ins
- MindTouch
  - Base platform with REST API
- Compiere
  - Core platform with drop-in jar files

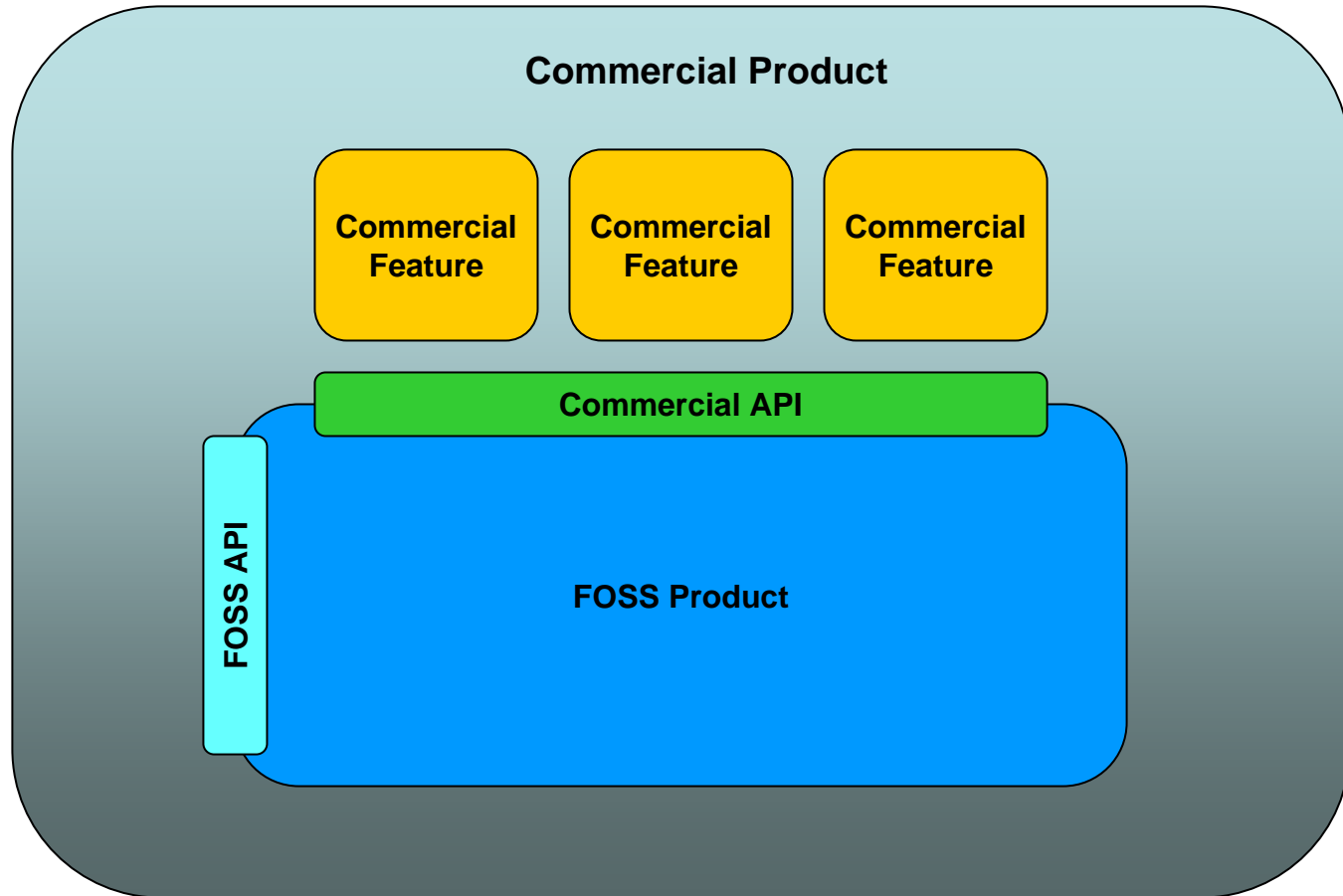
## Pros

- Same Binary
- Strict add-on to Open Source: no reinstall, re-test, etc.
- Allows targeting specific feature extensions

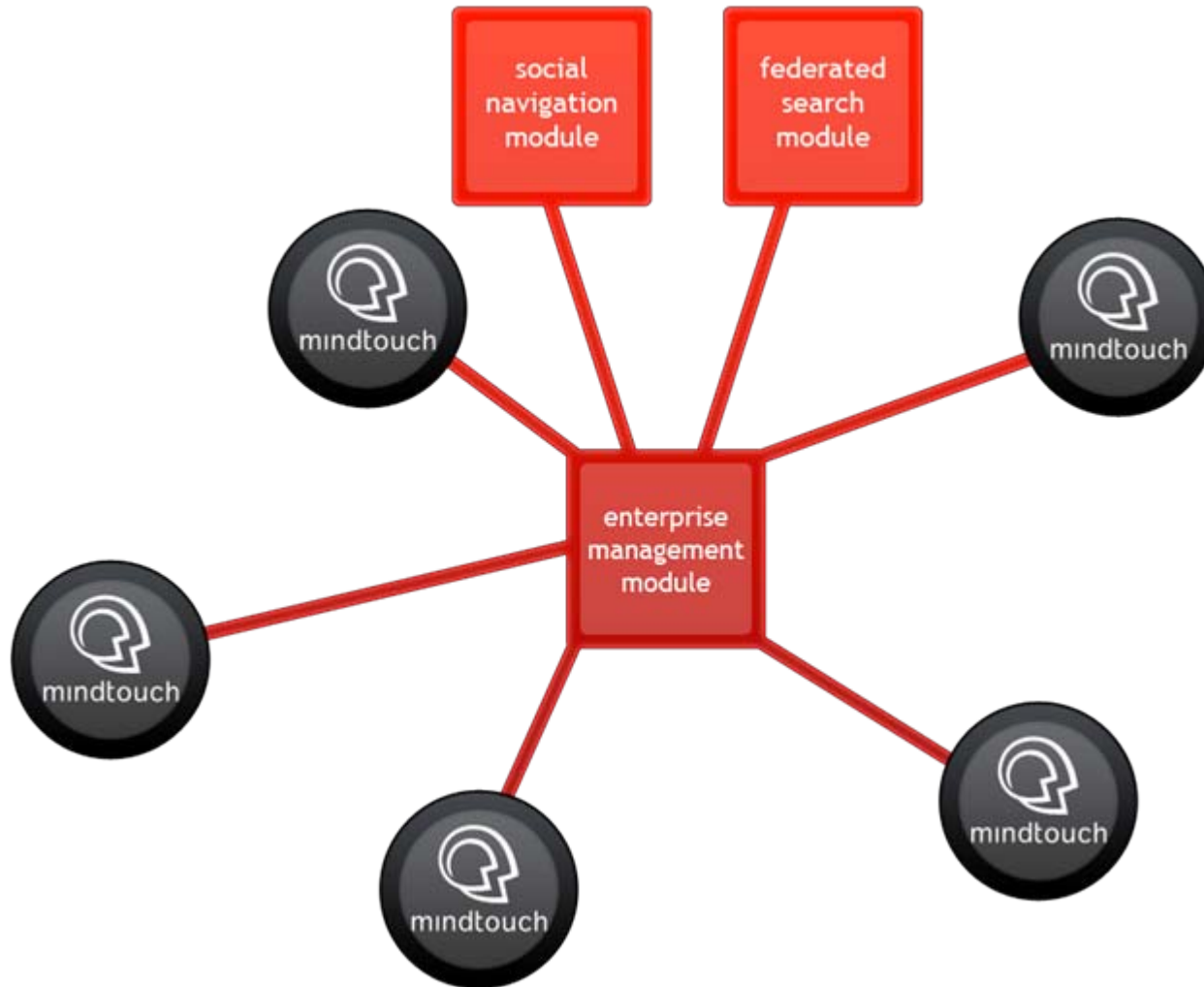
## Cons

- Third parties can add commercial extensions equally well
- Architecturally more difficult to implement

# DNN Commercial Module API



# MindTouch Federation



# Management

## Concept

- Give away the product and sell the tools to manage it to IT Operations
- Mission critical deployments need monitoring and management
- Open Source usage: more time than money
- Commercial usage: time, control, management at a premium

## Examples

- JBoss Operations Network (JON)
- MySQL Network
- MuleSource Network
- Sugar Network
- Pentaho Management Services

## Pros

- Strict add-on to Open Source: no reinstall, re-test, etc.
- Can be delivered as SaaS offering

## Cons

- Best when product is in mission-critical applications and any downtime is costly
- May compete/conflict with other management offerings

# Data Services

## Concept

- Give away the product and sell the data
- Many business applications need constantly updated reference data

## Examples

- Medsphere
  - Drug files need constant updates
  - Aggregate data on patient outcomes
  - Insurance codes/classifications
- Compiere
  - Tax tables (rates)

## Pros

- Same Binary
- Strict add-on to Open Source: no reinstall, re-test, etc.
- Requires ongoing subscription
- Can be delivered as SaaS offering

## Cons

- Data services are often resold: can be lower margin
- Utility/stickiness of open source offering is diminished without them

# Personalities

## Concept

- Same binary, different templates that create different “personalities”
- Drop in templates customize terminology
- Can be applied to existing installations (changes data, not code)

## Examples

- Compiere
  - Templates define different terminology, different look and feel for different industries
- DotNetNuke
  - Market place selling different look and feel (“skins”)
- Medsphere
  - Templates define different types of patient/doctor interactions

## Pros

- Same Binary
- Strict add-on to Open Source: no reinstall, re-test, etc.
- Allows targeting specific verticals

## Cons

- Not easily amenable to subscription offering
- Likely to compete with Open Source community
- Language packs are not good choice: tend to be community developed

# Rules/Workflows

## Concept

- Same binary, common rules or workflow engines
- Drop in different rules/workflows for different target markets

## Examples

- Compiere
  - Accounting rules specific to different geographies/industries
  - Tax codes, sales tax rules

## Pros

- Same Binary
- Strict add-on to Open Source: no reinstall, re-test, etc.
- Allows targeting specific markets
- Often changing: amenable to subscription offerings

## Cons

- Requires a more sophisticated architecture with embedded rules engine or workflow engine

# Automated Service Offerings

## Concept

- Automated service offerings that add value to the open source product
- Remote update service
- Migration services
- Computer based training
- Indemnification

## Examples

- Zend
  - CE -> apply patches manually
  - EE -> remote update service
- Compiere
  - CE -> manually migrate database to new versions
  - Automated update service

## Pros

- No changes to existing install

## Cons

- Create barriers to ongoing open source usage

# Summary

- Create an easy ramp from open source to commercial product
- Installation
  - Download is just the start
  - Think in terms of minutes from download to usable system
- Usage
  - Encourage usage of the installed product
- Contracting
  - Make the contracting process simple and forward looking
- Product architecture supports migration to commercial
  - Same Binary
    - Modularize
    - Management
    - Data Services
    - Personalities
    - Rules/Workflows
    - Automated Service Offerings

# Thank you.

**Larry M. Augustin**

[larry.augustin@gmail.com](mailto:larry.augustin@gmail.com)