The presentation discusses cloud computing business models, using open source business models as a point of comparison. Both cloud computing and open source are viewed as disruptive innovations that create new and enable old business models. I speculate about the future of cloud computing business models by looking at the history of open source business models.
Talk Contents

Open Source and Cloud Computing
Open Source Business Models
Cloud Computing Business Models
## Open Source and Cloud Computing Definition

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Legal definition</td>
<td>• A model for enabling</td>
</tr>
<tr>
<td>• Available in source code form</td>
<td>• ubiquitous,</td>
</tr>
<tr>
<td>• Rights to use, modify, distribute</td>
<td>• convenient,</td>
</tr>
<tr>
<td>• Realized by various licenses</td>
<td>• on-demand</td>
</tr>
<tr>
<td>• Process definition</td>
<td>• network access to</td>
</tr>
<tr>
<td>• Peer production</td>
<td>• a shared pool of</td>
</tr>
<tr>
<td>• Transparent process</td>
<td>• configurable computing resources</td>
</tr>
<tr>
<td>• Hinders vendor lock-in</td>
<td>• that can be</td>
</tr>
<tr>
<td></td>
<td>• rapidly provisioned and released</td>
</tr>
<tr>
<td></td>
<td>• with minimal management effort</td>
</tr>
<tr>
<td></td>
<td>• or service provider interaction.</td>
</tr>
</tbody>
</table>

While open source is well understood, cloud computing is not. Open source has a legal and pragmatic (process) definition, see left side of slide, as defined by the Open Source Initiative.

Cloud computing typically means a delivery model of software or hardware (“as a service”) that is fine-grain, i.e. can happen in small increments and that is priced accordingly in small increments (“pay-as-you-go”). All of this when compared to the prior state of the art which was not as fine-grain.
A **business model** describes the rationale of how an organization creates, delivers, and captures value.[1]
The business model canvas is a tool for analyzing business models. One strength is the visual depiction and handiness.
Types of Business Models in this Context

1. Innovation
2. Enablement
3. Support

In both open source and cloud computing we can distinguish two types of business models: Native open source and cloud computing business models and those in support of it, but that are not open source or cloud computing business models in themselves. Within the native business models we can distinguish, pragmatically, those that are new and those that have been revamped from prior existing models.
<table>
<thead>
<tr>
<th>Innovation</th>
<th>Enablement</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Sales and marketing</td>
<td>• Sales and marketing</td>
</tr>
<tr>
<td>• Growing the actual market</td>
<td>• Growth in existing market</td>
</tr>
<tr>
<td>• Customer land grab</td>
<td>• Customer acquisition harder</td>
</tr>
<tr>
<td>• Prod. mgmt. and engineering</td>
<td>• Prod. mgmt. and engineering</td>
</tr>
<tr>
<td>• Unclear requirements</td>
<td>• Understood requirements</td>
</tr>
<tr>
<td>• Experimentation, pivoting</td>
<td>• Execution</td>
</tr>
</tbody>
</table>

The distinction between innovation and enablement is pragmatic and derived from the market companies in the respective space are facing:

Newly invented business models face a market where the companies pursuing the model still have to educate their customers as to their needs (for this product or service). It is a land growth and grab situation.

Enabled business models face a market where customers already understand the value proposition, however, because of existing rivals, you may have “to pry your customers from your competitor's cramped hands”.
Example Businesses (and Models)

Open Source

- Innovation examples
  - Distributor models: Red Hat, SUSE, Eucalyptus
- Enablement examples
  - Open core: SugarCRM, Alfresco
  - Pure service models
- Support examples
  - Black Duck, Palamida

Cloud Computing

- Innovation examples
  - SaaS: iCloud, Dropbox, Pandora
  - PaaS: App Engine, Force.com
  - IaaS: Amazon EC2
- Enablement examples
  - SaaS: SugarCRM, Salesforce
- Support examples
  - Eucalyptus, Red Hat, VMware
  - Hardware: IBM, Dell

In open source, distributor business models are novel. Open core business models compete more directly with established firms and have been derived from traditional closed source models.

Support business models aren't open source or cloud computing business models themselves but support those.
Open source is not a business model.

Open source is a distribution and development methodology.

By now it is well established that open source is not a business model but rather a distribution (read: sales and marketing) and a development (read: product management and engineering) strategy.
Open source is not a business model.

Open source is a distribution and development methodology.
Working assumption:

**Cloud computing is not a business model.**

**Cloud computing is (mostly) a distribution methodology.**

I'm assuming then that cloud computing is not a business model itself but rather an enabler of existing and new business models. This enablement is mostly a sales and marketing strategy, and somewhat less important, an engineering (cost reduction) strategy. Cost reduction only kicks in if a cloud provider bases its offering on some other xaaS.
Cloud computing providers rely on other cloud computing providers, e.g. SaaS relies on PaaS relies on IaaS so we can see a consumer/supplier relationship repeating through the technology stack.

Sales benefits trickle down and cost benefits bubble up the cloud computing stack.
So... sales and marketing. Open source created a new and novel sales funnel. The biggest hurdle that remains is that someone inside a (potential customer) company still had to install the software, i.e. technical expertise was required, even if the software was just running on a desktop machine somewhere in some business unit.

Cloud computing removes that problem by making it easy to consume a service over the network. No need to install software (though you still have to configure it). This does away with a major consumption problem. The possibly required price for a single user is typically neglectable. The result is a much smoother user-driven adoption of a product. IT and purchasing is ignored, at least initially.
With a foot in the door, like in open source, cloud computing providers can organize their free or minimally-paying users to achieve a real enterprise sale.
Open source provides various benefits for business development, product management, and engineering. These are reasonably well understood.

One might expect some of these for cloud computing as well, because of the simplified consumption of products and services, but the jury is still out on this. The only obvious benefit is lower costs if a cloud computing provider can itself rely on lower-layer cloud computing providers.

<table>
<thead>
<tr>
<th>Official Definition</th>
<th>What's Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Open formats and interfaces</td>
<td></td>
</tr>
<tr>
<td>• Make all data available</td>
<td></td>
</tr>
<tr>
<td>• Must follow open standard</td>
<td></td>
</tr>
<tr>
<td>• Open standards</td>
<td></td>
</tr>
<tr>
<td>• IP exclusion rights excluded</td>
<td></td>
</tr>
<tr>
<td>• Open source implementation</td>
<td></td>
</tr>
<tr>
<td>• Needs open process</td>
<td></td>
</tr>
<tr>
<td>• Leads to agreed-upon standards</td>
<td></td>
</tr>
<tr>
<td>• Mentioned, but not required</td>
<td></td>
</tr>
<tr>
<td>• Learning from the ASF</td>
<td></td>
</tr>
<tr>
<td>• Multiple independent participants</td>
<td></td>
</tr>
<tr>
<td>• Competing projects</td>
<td></td>
</tr>
</tbody>
</table>

"Every real-world specification is an underspecification."


The recent definition of open cloud principles suffers from not requiring a consensus process for the definition of open standards. Here, open source foundations like the Apache Software Foundation can lead the way. For example, open standards should only be considered as such if there are at least three independent parties involved in their standardization and have been at it for a sustained period of time. Otherwise, the standard should not leave the incubating stage.
More Typing of Business Models

Open Source

- Open source distributor
  - Does not own source code
  - Provides commoditized software
  - Differentiated by service quality

- Single-vendor open source firm
  - A.k.a. “open core”, “dual licensing”
  - Owns and maintains source code
  - Differentiated by software

Cloud Computing

- Utility computing
  - Standardized software stack
  - Differentiated by service quality
  - Adheres to Open Cloud Principles

- Single-source cloud computing
  - Uses proprietary software to differentiate offering + service
  - Adheres to Open Cloud Principles only in name; locks in
Like in open source, I expect cloud computing to have an innovation frontier, continuously being pushed outward by a trailing commoditization frontier.
This canvas shows the business model benefits for a generic open source software product firm.
This canvas shows the benefits of cloud computing for a generic xaaS provider.

Please note the absence of “community”, a key enabling factor for open source business models. It remains to be seen, as noted earlier, whether through the simplified consumption provided by cloud computing providers equivalent effects can be created.
SugarCRM provides CRM software. It has both an open source and cloud computing strategy. Only half as old as Salesforce, it has grown significantly because of this strategy. In particular open source helped finding resellers in non-US jurisdictions that are hosting SugarCRM where Salesforce had to build up their own local subsidiary. (Many countries and customers won't allow critical data to leave country boundaries.)
Conclusions

Cloud computing...

• is not a business model
• but an enabler of new models
• with open source synergies
Thank you! Questions?

dirk@riehle.org – http://dirkriehle.com – @dirkriehle

dirk.riehle@cs.fau.de – http://osr.cs.fau.de