Open Source professionell einsetzen

Mein Background

- Ich bin überzeugt von Open Source.
- Ich verwende fast nur Open Source privat und beruflich.
- Ich arbeite seit mehr als 10 Jahren mit Linux und Open Source.
- Ich arbeite seit mehr als 10 Jahren in der Telekommunikation und IT.

Was bedeutet Open Source?

- Open Source bedeutet freier Zugang zum Quellcode.
- Das Recht diesen zu verändern, weiter zu entwickeln, zu vervielfältigen, weiter zu geben und beliebig zu benutzen.
- Dieses Recht wird mittels Open Source Lizenzen geschützt, welche auch Pflichten mit sich bringen.
- Open Source wird von und für die Allgemeinheit geschaffen.

Unterschiede zwischen Open Source und Kommerzieller Software

	Open Source	Kommerzielle Software
Garantie / Gewährleistung	Nein	Ja
Strukturiertes Management	Nein	Ja
Vertraglich gesicherter Support	Nein	Ja
Lizenzkosten	Nein	Ja
Source Code	Ja	Nein
Rechte am Source Code	Ja	Nein
....		

Ausgangslage für die Tabelle ist das man das jeweilige Open Source Produkt nicht über einen Dienstleister erwirbt.

Warum bzw. warum nicht viele Open Source einsetzen wollen

Warum Open Source?

Kosten sparen

Warum nicht Open Source?

größeres Risiko

Kosten?

Open Source hat keine Lizenzkosten, aber man muss trotzdem Geld investieren um Open Source erfolgreich und professionell einzusetzen.

Open Source vs. Kommerzielle Software – wie ich es sehe

Open Source = Freiheit

Freiheit = Sicherheit

Dadurch man den Quellcode hat, ist man ein Teil des Produkts.

Kommerzielle Software = Abhängigkeit

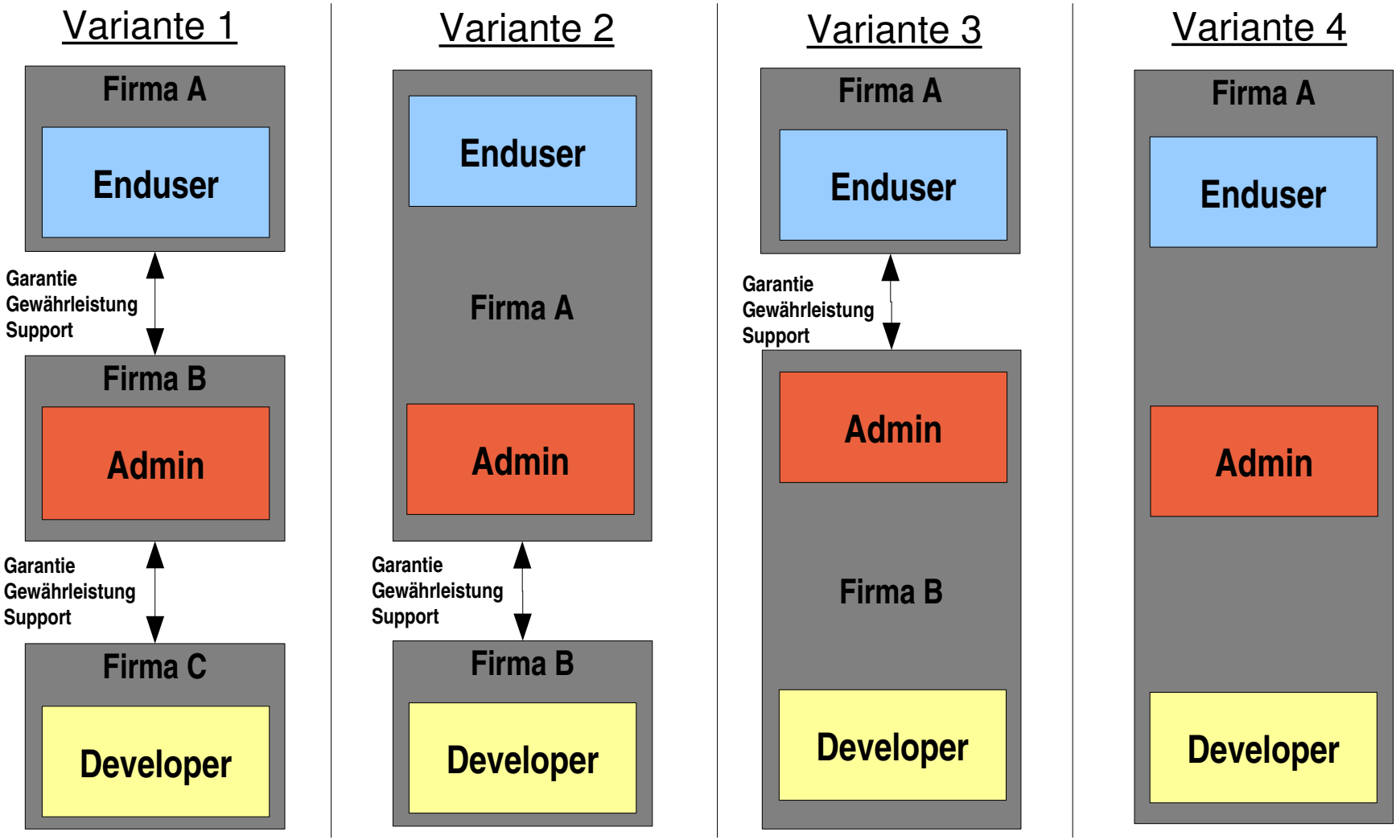
Abhängigkeit = Risiko

Ohne den Quellcode zu besitzen ist man immer abhängig vom Hersteller.

Es gibt 3 Gruppen im professionellen Bereich

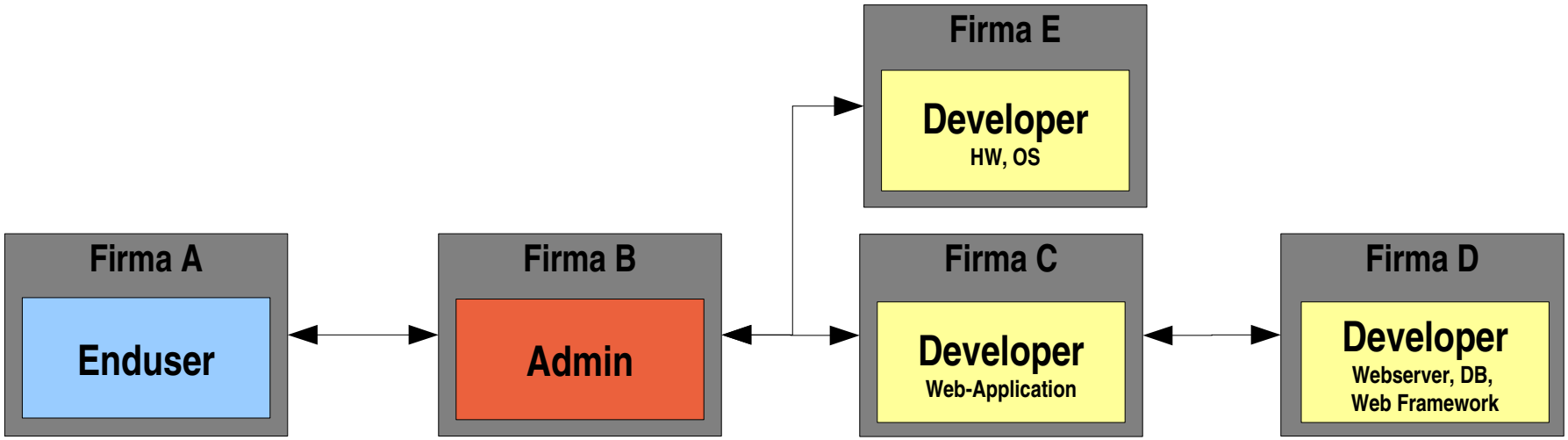
- Enduser
  - Verwendet ausschließlich Software
- Administrator
  - Installieren, integrieren, konfigurieren und Support
- Developer
  - Entwickelt Software

Klassische Szenarien beim Einsatz kommerzieller Software



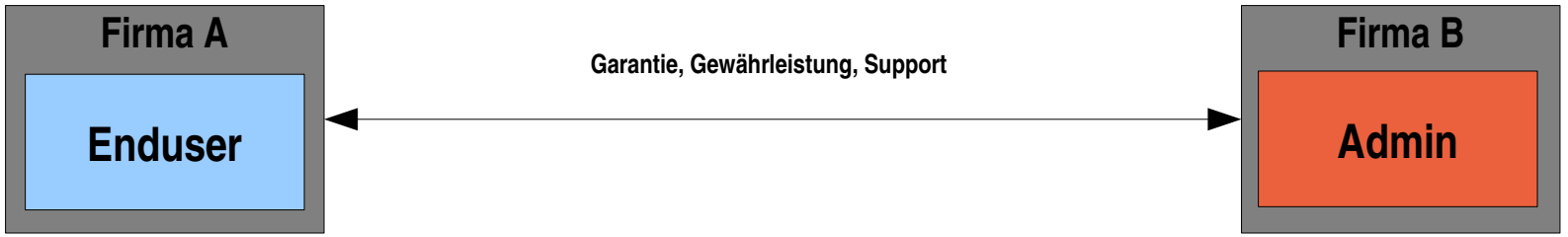
Bsp. Web basierende Zeiterfassung

- Enduser
  - Verwendet ausschließlich die Applikation, z.B. drückt im Intranet „kommen“ und „gehen“
- Administrator
  - Warten die Server (Hardware und OS) und die Applikation (Webserver, DB, Web-Applikation)
- Developer der Web-Applikation
  - Setzt auf Webserver, DB und Web Framework von anderen Herstellern auf und gibt HW und OS vor.



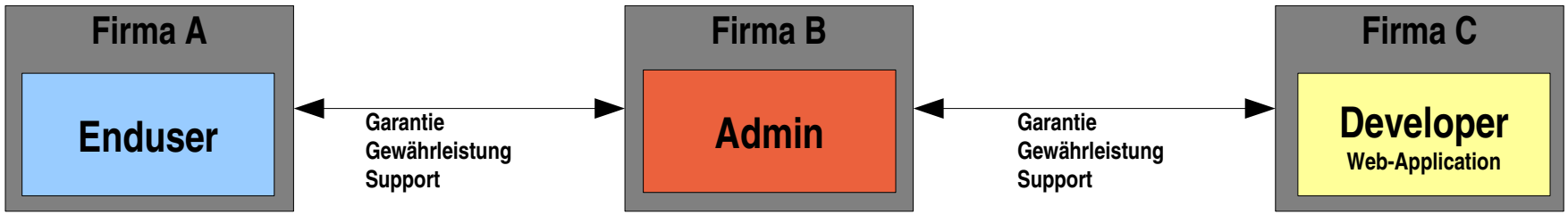
Enduser erwirbt Software

- Für den Enduser entsteht so gut wie kein Unterschied, ob es sich um Open Source handelt oder nicht.
- Der Enduser im professionellen Bereich hat immer eine Admin Gruppe als Schnittstelle. Muss sich der Enduser die Applikation selber administrieren wäre er Teil der Admin Gruppe.
- Garantie, Gewährleistung und Support wird von Firma B (IT-Dienstleister Admin Gruppe) übernommen.
- Open Source:
  - Durch das Verwenden von Open Source bin ich nicht direkt vom IT-Dienstleister/Hersteller abhängig
  - Oft günstiger



Szenario 1 - Firma B erwirbt kommerzielle Software

- Nachteile:
  - Abhängigkeit vom Hersteller
  - Anschaffungs- und Lizenzkosten
- Vorteile:
  - Geschlossene Support-kette
  - Vertragliche Sicherheit (Garantie, Gewährleistung und Support)



Abhängigkeit vom Hersteller

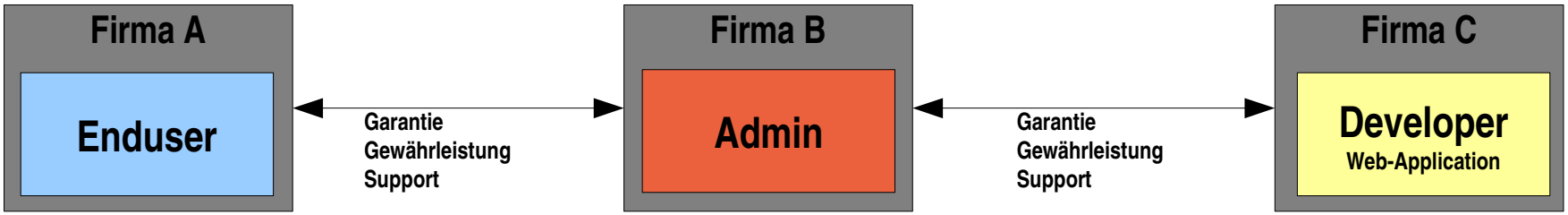
Da man nicht im Besitz des Source Codes ist, ist man abhängig vom Hersteller

- Hersteller geht Konkurs
  - Geht der Hersteller in Konkurs sind alle rechtlichen Sicherheiten hinfällig.
- Request for Feature
  - Liegen von mir gewünschte Funktionserweiterungen nicht auf der Roadmap oder im Interesse des Herstellers, kann eine Implementation sehr teuer oder sogar unmöglich sein.
- Support
  - Oft ist der Kunde nicht König sondern Bittsteller.

Beim Einsatz von Open Source entsteht keine direkte Abhängigkeit, da man die Rechte am Source Code besitzt.

Szenario 2 - Firma B erwirbt Open Source Software (Enterprise Version)

- Nachteile:
  - Anschaffungskosten
  - Aber auch keine direkten Vorteile von dem Open Source Prinzip, da mir so gut wie immer nur die vom Vertragspartner erhaltene Release supported wird.
- Vorteile:
  - Geschlossene Support-kette
  - Durch das Verwenden von Open Source bin ich nicht direkt vom Hersteller abhängig
  - Oft günstiger



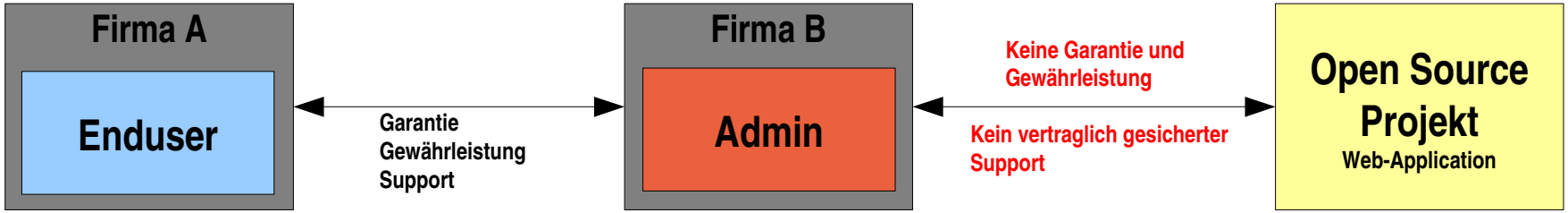
Enterprise Version

Enterprise Version bedeutet meistens, dass man Open Source bei einer Firma kauft. Dadurch bekommt man Garantie, Gewährleistung und Support.

Man bekommt diese aber nur auf eine dezidierte Version.

Szenario 3 - Firma B verwendet Open Source Software (Community Version)

- Nachteile:
  - Keine geschlossene Support-kette
  - Ich muss Garantien, Gewährleistung und Support auf Produkte geben, die ich nicht entwickelt habe.
- Vorteile:
  - Keine Software Anschaffungskosten
  - Keine Abhängigkeit von einem Lieferanten
- Vor- oder Nachteile?
  - Verantwortung liegt bei mir, man muss sich um einiges kümmern, man kann mehr selbst bestimmen und man ist Teil des Produkts.



Tipps zum Risiko minimieren mit Community Version

- Die richtige Strategie ist entscheidend!
- Know-How ist die beste Sicherheit
  - Nicht Unbekannten blind vertrauen.
  - Erfahrungsaustausch im nahen Umfeld.
  - Erfahrungsaustausch in der Community.
  - Testen, testen, testen...
  - Binary oder selbst kompilieren?
  - Nicht jedes Open Source Projekt ist auch für den professionellen Einsatz gedacht.
  - Manche Projekte verschwinden so schnell wie sie entstehen.
  - Gibt es eine Community (Doku, Support forum, viele google treffer,...)
  - Lebt das Projekt noch (neue Releases, neue Features, Bug fixes,...)
  - ….

Open Source JA oder NEIN

Die meisten Firmen überzeugt nur der Business Case.

Risiko?

Abhängigkeiten können das größte Risiko darstellen, da sie sehr hohe Kosten erzeugen können.

Open Source vs. Kommerzielle Software – wie ich es sehe

Open Source = Freiheit

Freiheit = Sicherheit

Dadurch man den Quellcode hat, ist man ein Teil des Produkts.

Kommerzielle Software = Abhängigkeit

Abhängigkeit = Risiko

Ohne den Quellcode zu besitzen ist man immer abhängig vom Hersteller.

Fragen?

albert@hayr.at

www.hayr.at