

Konfiguration unter Linux mit Elektra

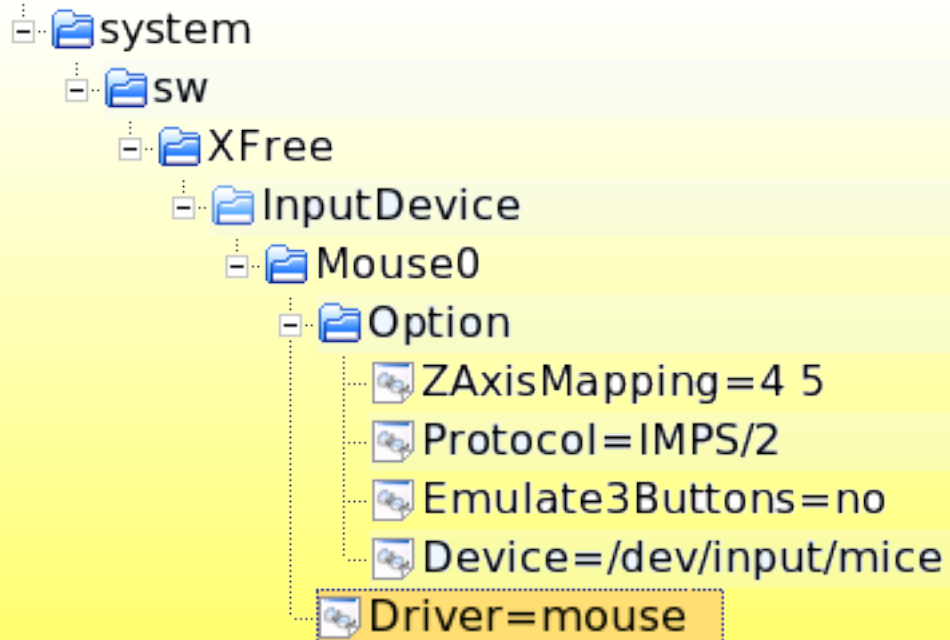
Linuxwochen Wien 2010

Markus Raab <elektra@markus-raab.org>

<http://www.libelektra.org>



Was ist Konfiguration?



```
Section "InputDevice"
  Identifier "Mouse0"
  Driver "mouse"
  Option "Protocol" "IMPS/2"
  Option "Device" "/dev/input/mice"
  Option "ZAxisMapping" "4 5"
  Option "Emulate3Buttons" "no"
EndSection
```

✓ Einstellungen und Optionen von Programmen

✓ Einfache hierarchische Struktur

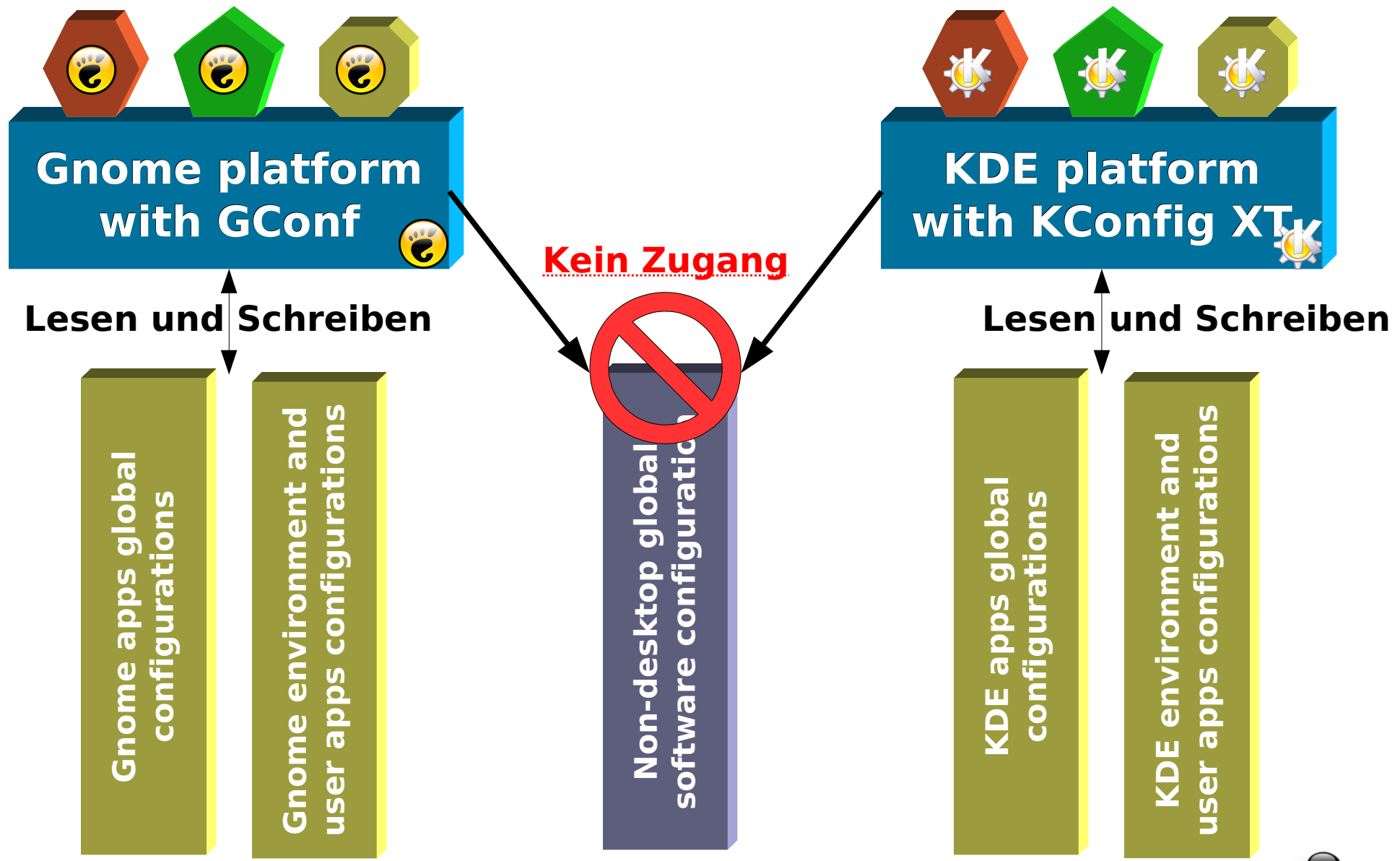
✓ Für Menschen lesbar

✓ Für Menschen editierbar

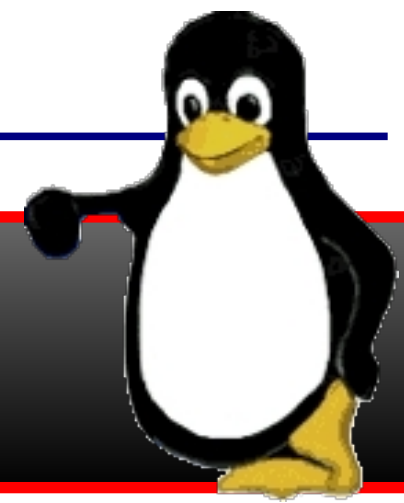
✗ Nicht für alle Programme einfach lesbar/editierbar



Konfiguration für jede Plattform?



Was ist Elektra?



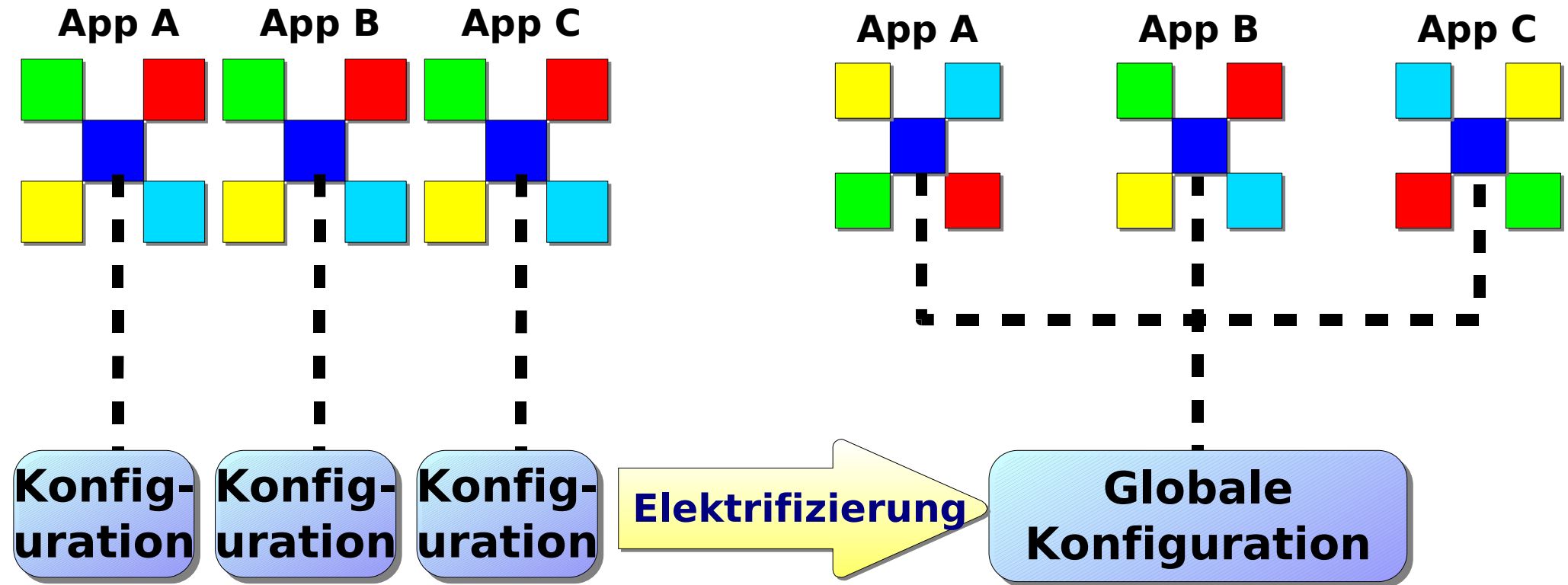
```
bash$ ldd /lib/libelektra.so
libc.so.6 => /lib/tls/libc.so.6
/lib/ld-linux.so.2 => /lib/ld-linux.so.2
```

- ✓ *Portable Bibliothek*
- ✓ *Lese- und Schreibzugriff auf abstrakte Konfiguration*
- ✓ *Sicherheit über OS*
- ✓ *Pragmatische Sicht*
 - ✓ *Konfigurationsdateien*
 - ✓ *Applikationsspezifische Eigenheiten*

<http://www.libelektra.org>



Elektrifizierung

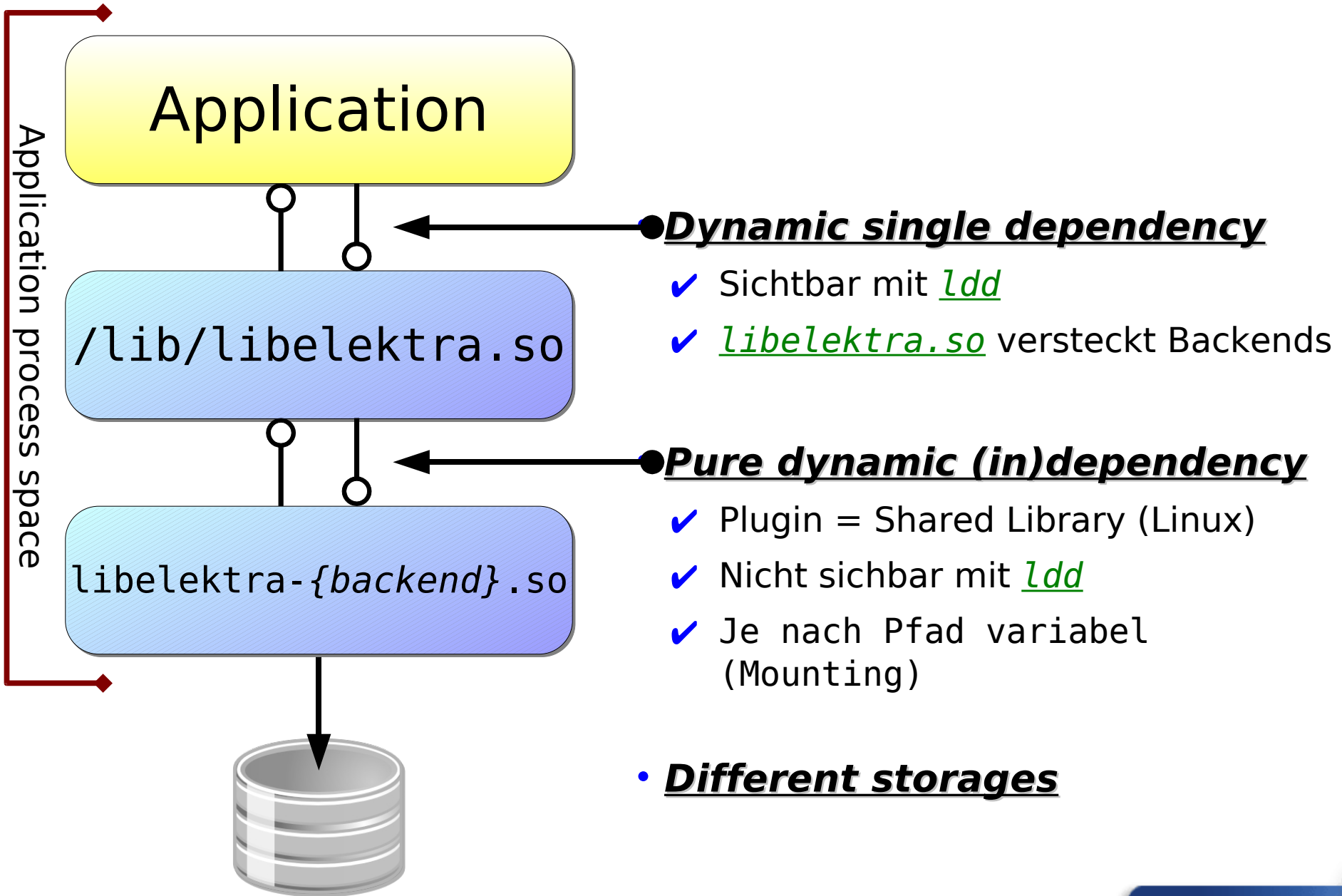


- ✗ Konfiguration ist stark abhängig von der Applikation
- ✗ Konfiguration wird nicht mit anderer Software geteilt

- ✓ Konfiguration repräsentiert durch Standards
- ✓ Verschiedene Programme können Konfigurationen teilen



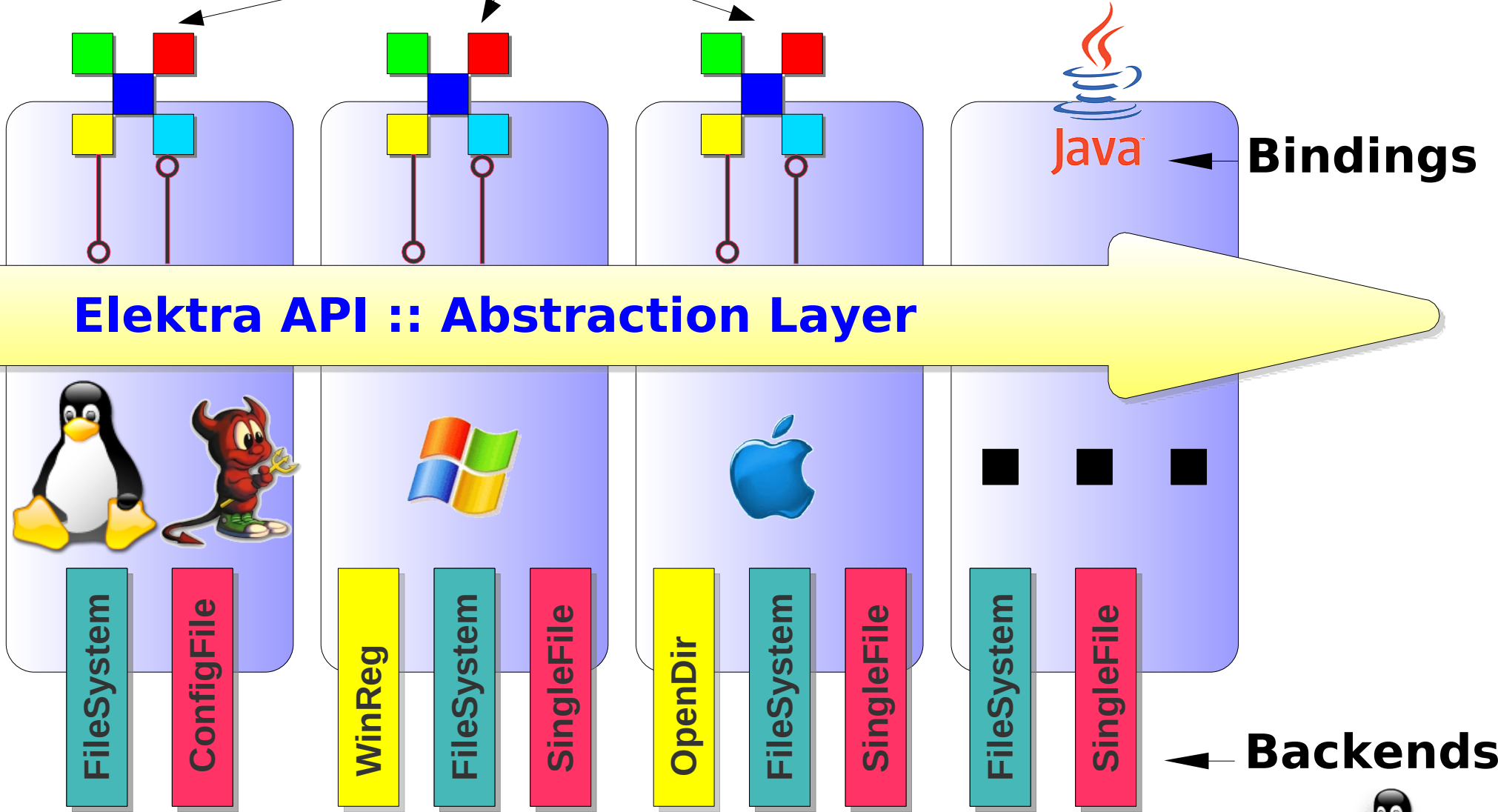
Technische Umsetzung



Was ist das Ziel?

Cross and Non-Cross Platform Software

Java, Apache, Samba, KDE, /sbin/init, ...



<http://www.libelektra.org>



Key Database Semantik






Linuxwochen Wien 2010

Markus Raab <elektra@markus-raab.org>

<http://www.libelektra.org>



Key Hierarchie

 system	key/value pairs für System
 user	key/value pairs für aktuellen User
 user:aviram	Avi's persönliche keys
 user:luciana	Luciana's persönliche keys
 user:valeria	Valeria's persönliche keys

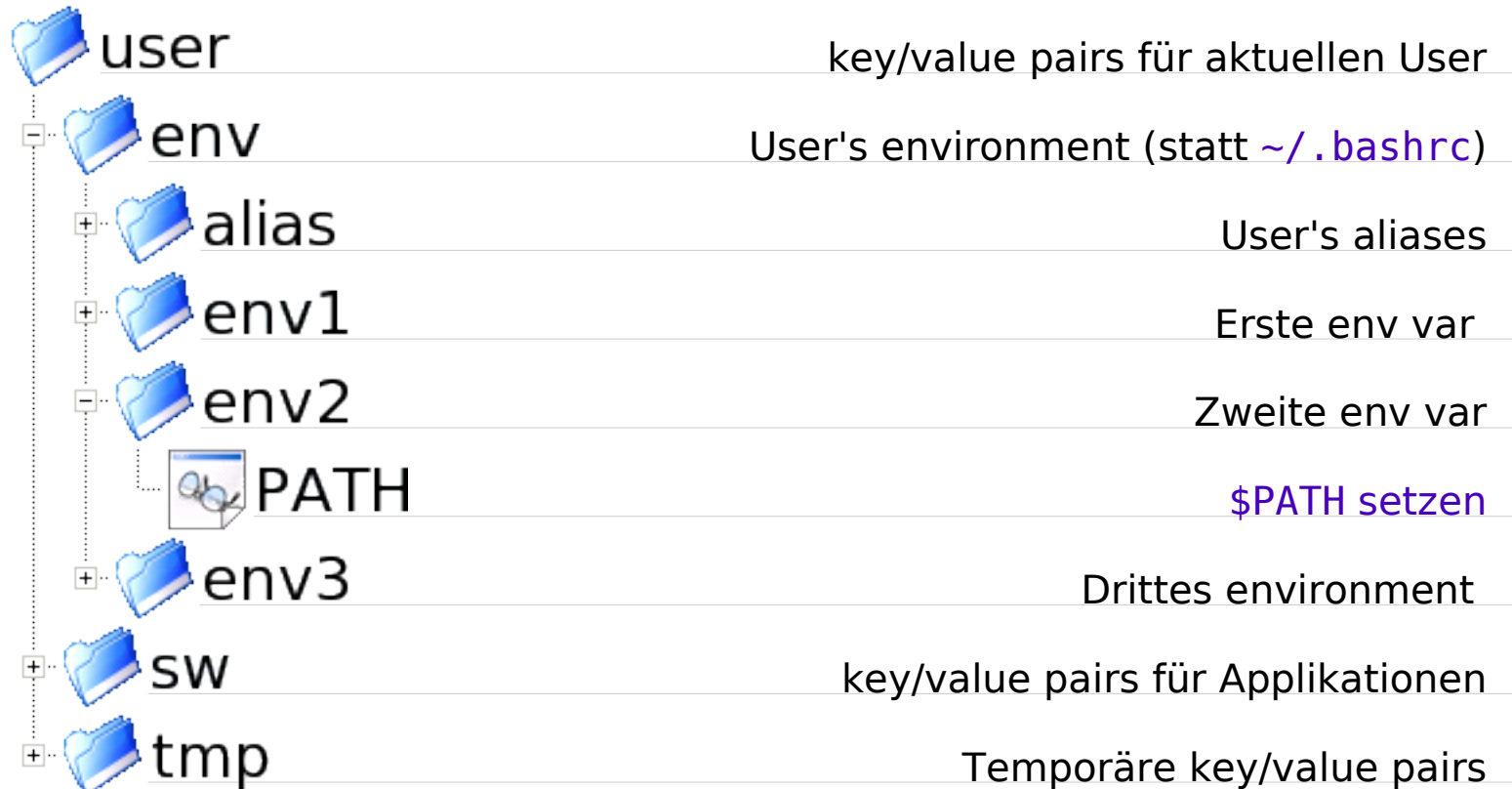
- ◆ Die system/* Hierarchie ist gespeichert unter /etc/kdb/
- ◆ Die user:\$USER/* Hierarchie unter ~\$USER/.kdb/
- ◆ Die user/* Hierarchie ist eine Abkürzung für aktuellen User.



Key Hierarchie :: *system/...*

 system	key/value pairs für System
 filesystems	Gleichwertig zu /etc/fstab
 groups	Gleichwertig zu /etc/group
 hw	Statische gefundene Hardware
 init	Gleichwertig zu /etc/inittab
 network	Netzwerk Konfiguration
 SW	Applikationsspezifisches
 regedit	Applikation 1
 XFree	Applikation 2
 users	Gleichwertig zu /etc/passwd

Key Hierarchie :: *user/...*



Kaskading

Beispiel für Key Name

Value

system/filesystems/boot/device

/dev/hda1

/shortcut/menu/open

Strg+O

system/net/TSP/ANL/phone

555-2132

**Durch diesen Slash wird
der richtige Schlüssel
automatisch gesucht**

user@msn.com

Driver

radeon

system/sw/httpd/site1.com/DocRoot

/var/www/site1.com

user/env/alias/ls

ls -Fh

user:valeria/sw/regedit/gui/width

747

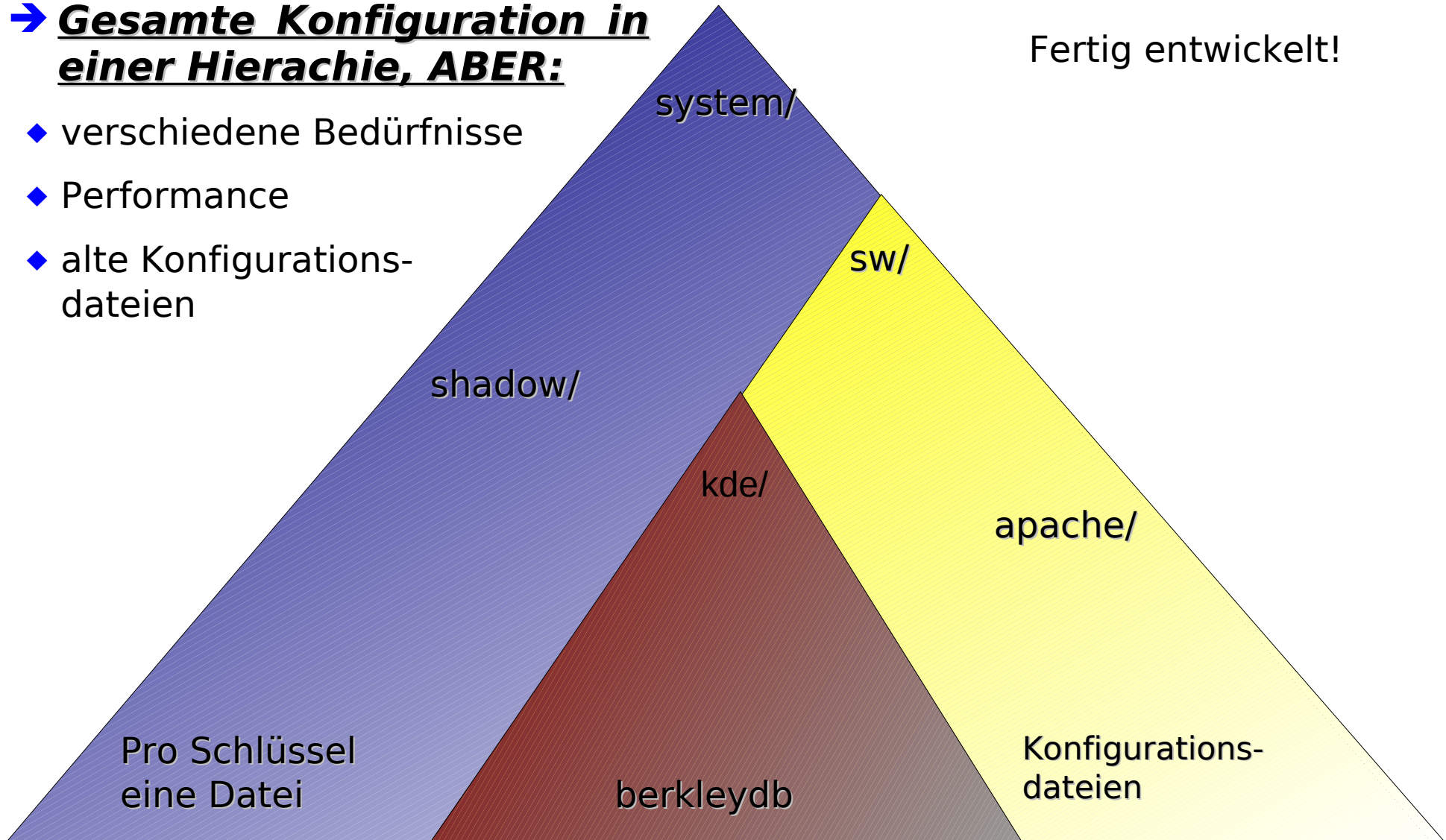


Backends mounten

→ Gesamte Konfiguration in einer Hierarchie, ABER:

- ◆ verschiedene Bedürfnisse
- ◆ Performance
- ◆ alte Konfigurationsdateien

Fertig entwickelt!



Warum Mounting?

```
#inetd.conf
Ident    stream  tcp      wait    identd  /usr/sbin/identd  identd
saft     stream  tcp      nowait  root    /usr/sbin/tcpd    /usr/sbin/sendfiled
finger  stream  tcp      nowait  nobody  /usr/sbin/tcpd    in.fingerd
```

```
service saft
{
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    server = /usr/sbin/sendfiled
    port = 487
    disable = no
}
```

- ✓ Unterschiedliche Bedürfnisse
- ✓ Identität von Software

Administration mit Elektra

Linuxwochen Wien 2010

Markus Raab <elektra@markus-raab.org>

<http://www.libelektra.org>



The *kdb* Command: Perfekt für Skripte

Manual + *kdb --help* vorhanden!

```
$ kdb get system/filesystems/boot/mpoint  
  
# kdb set system/sw/xorg/Screen/Display/Modes 1280x1024  
  
$ kdb ls -Rv system/sw/myapp  
  
$ kdb rm user/env/alias/vnc  
  
$ kdb info user/  
/home/markus/.kdb/user
```



XML Import and Export



```
$ kdb export user/env/alias > file.xml
$ kdb import file.xml
```

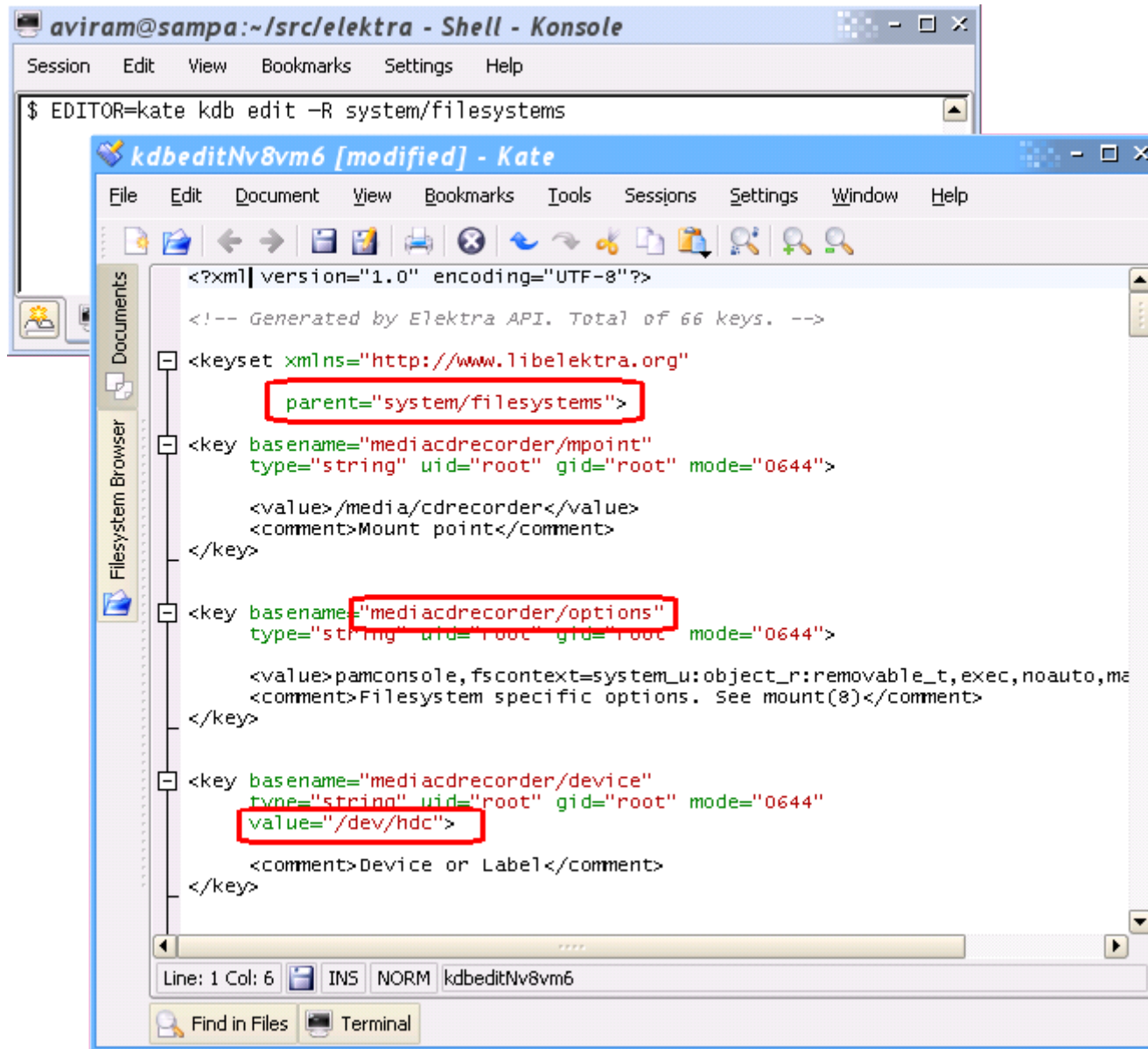
```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Output of a 'kdb export user/env/alias' command -->

<keyset xmlns="http://www.libelektra.org"
  xsi:schemaLocation="http://www.libelektra.org elektra.xsd"
  parent="user/env/alias">
  <key basename="ls"
    type="string" uid="aviram" gid="aviram" mode="0664"
    value="ls -Fh --color=auto"
    <comment>Make 'ls' command more clever</comment>
  </key>
  <key basename="vnc"
    type="string" uid="aviram" gid="aviram" mode="0664">
    <value>vncserver -geometry 900x650</value>
    <comment>Instant creation of a VNC server</comment>
  </key>
</keyset>
```



Editieren von key/value pairs

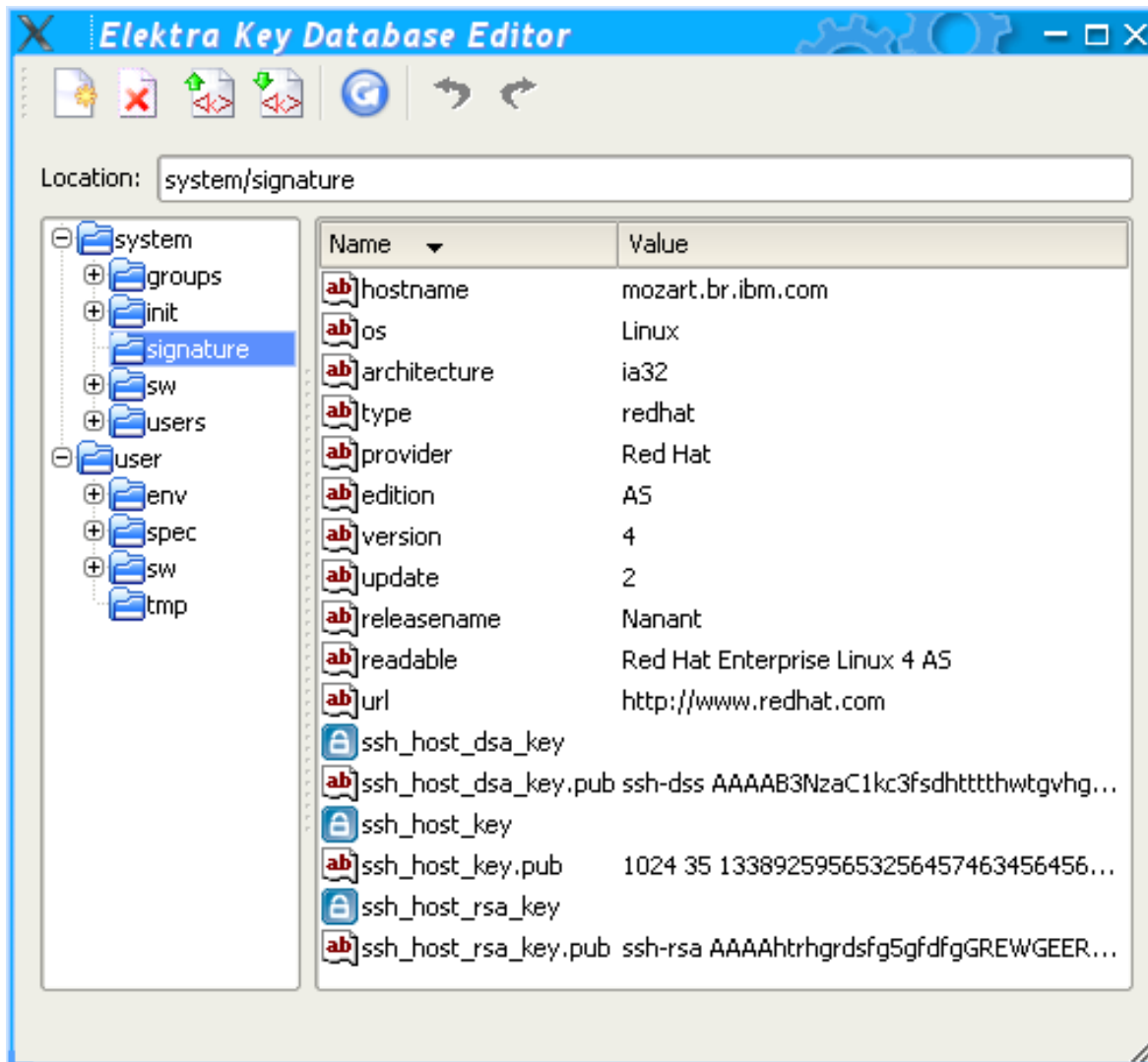


The image shows a terminal window and a Kate editor window. The terminal window displays the command `$ EDITOR=kate kdb edit -R system/filesystems`. The Kate editor window shows the XML content of the file `kdbeditNv8vm6`. The XML content is as follows:

```
<?xml| version="1.0" encoding="UTF-8"?>
<!-- Generated by Elektra API. Total of 66 keys. -->
<keyset xmlns="http://www.libelektra.org"
  parent="system/filesystems">
  <key basename="mediacdrecorder/mpoint"
    type="string" uid="root" gid="root" mode="0644">
    <value>/media/cdrecorder</value>
    <comment>Mount point</comment>
  </key>
  <key basename="mediacdrecorder/options"
    type="string" uid="root" gid="root" mode="0644">
    <value>pamconsole, fscontext=system_u:object_r:removable_t, exec, noauto, me
    <comment>Filesystem specific options. See mount(8)</comment>
  </key>
  <key basename="mediacdrecorder/device"
    type="string" uid="root" gid="root" mode="0644"
    value="/dev/hdc">
    <comment>Device or Label</comment>
  </key>
```

- Beliebiger Editor um XML Datei zu editieren
- Neu hinzugefügte werden, kommen **hinzu.**
- Veränderte werden **übertragen.**
- Auch **löschen** funktioniert.

kbedit: The Elektra GUI Edit Tool



- auch für Beginner
- Hierarchische Struktur
- Alle Funktionen
- Exportieren und Importieren

Wenn alles versagt hast du hoffentlich das richtige Backend gewählt.

Demo

- *Auf Nokia N900*
- *ARM Cortex 600 Mhz*
- *256 MB Ram*
- *Interner Flash mit ca. 15 MB/sec*

Modulare Backends für Elektra

Linuxwochen Wien 2010

Markus Raab <elektra@markus-raab.org>

<http://www.libelektra.org>



Key Namen

<u>Beispiel für Key Name</u>	<u>Value</u>
system/filesystems/boot/device	/dev/hda1
system/net/resolver/server	192.168.10.1
system/net/ISP/AOL/phone	555-2132
system/net/ISP/.MSN/login	user@msn.com
system/sw/XFree/Device/Videocard0/Driver	radeon
system/sw/httpd	/var/www/site1.com
user/env/alias	ls -Fh
user:valeria/sw/regedit/gui/width	747

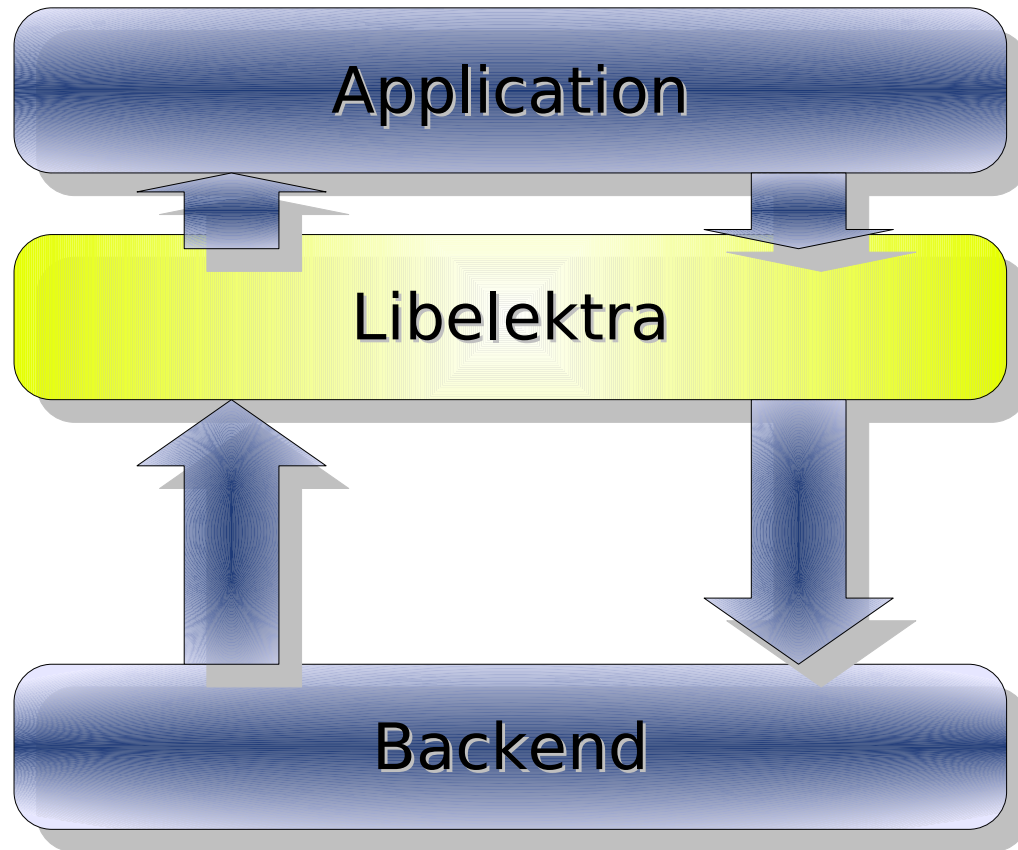
**Dieser kleiner Punkt
macht die ganze
Hierarchie inaktiv?**



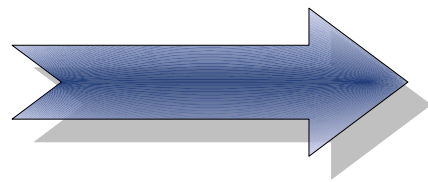
KDB Semantik (Not final) aka 7 REGEL

1. Namensräume: User und System
2. Keys sind mit Namen **eindeutig** definiert
3. *Löcher* werden akzeptiert
4. Keine Ordnungsrelation (Set)
5. Keys sind unabhängig
6. Keine Links, Versteckte Keys o.ä. FS-Semantiken
7. Aber Plugins und Metadaten können 3-6 abändern

Derzeitige Situation



- *Duplizierter Code*
- *Fehlende Flexibilität*
- *Reimplementierung von Features*
- *Capabilities*

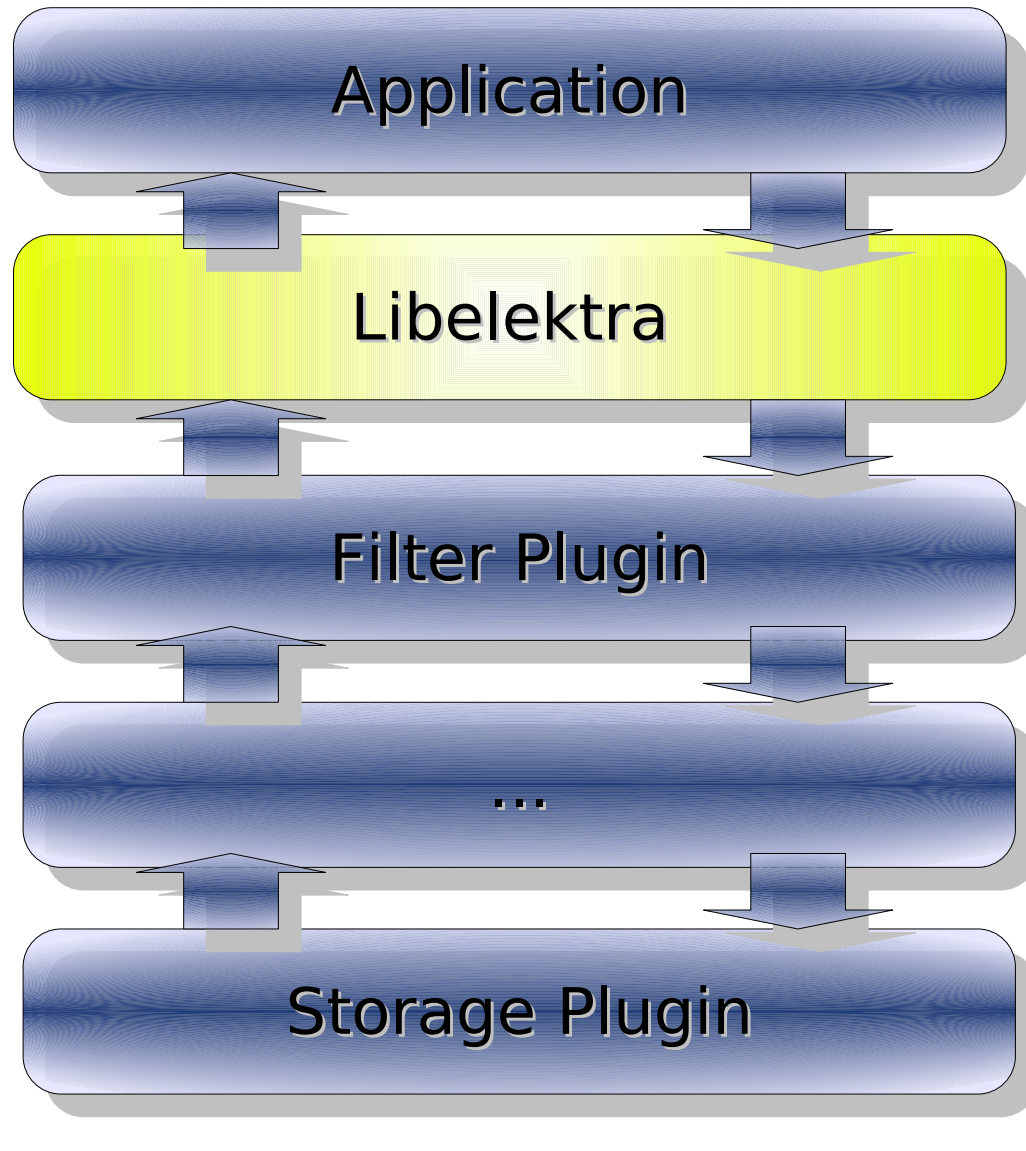


Modularität in
Backends?

<http://www.libelektra.org>



Erwünschte Situation



- *Separation of Concerns*
- *Unix Philosophie*
- *Flexibles Laufzeitverhalten*

n Plugins = 1 Backend

Elektra als Bibliothek



- *Bücher: Key (Titel/Inhalt)*
- *Regale: KeySet*
- *Bibliothekar: Backend*

€ e Ll € e Kt Tt Kr Aa

<http://www.libelektra.org>



Serialization / Storage Plugins



➤ *Eet*

➤ *XML*



➤ *Ini*

➤ *C*



➤ *Lua*

➤ *Tcl*

```
<?xml version="1.0"
encoding="UTF-8"?>

<keyset xmlns=""
xsi:schemaLocation=""

parent="user/env/alias">
  <key basename="ls"
type="string"
value="ls -Fh"

  <comment></comment>
</key>

  <key basename="vnc"
type="string">
  <value>vncserver
-geometry
900x650</value>
  <comment></comment>
</key>
</keyset>
```

```
ksNew( 17 ,
  keyNew
    ("user/fstab/rootfs",
KEY_TYPE,
KEY_TYPE_STRING,
KEY_END),

  keyNew
    ("user/fstab/rootfs/device",
KEY_VALUE, "LABEL=/",
KEY_TYPE, KEY_TYPE_STRING,
KEY_END),

  keyNew
    ("user/fstab/rootfs/type",
KEY_VALUE, "ext3",
KEY_TYPE, KEY_TYPE_STRING,
KEY_END),
KS_END);
```

```
{
  {system/hosts ""
  {file /etc/hosts
  mode 644
  mtime 1273165467}
}

{system/hosts/dns1
213.129.232.1
{comment "my first dns"
text_before "#hosts"
order 1}
}

{system/hosts/dns2
213.129.232.1
{comment "my second dns"
text_after "#line"
order 2}
}
}
```



Anwendung von Plugins (1)

✓ Encoding

```
$ kdb get user/sw/soffice/userdata/name  
Could not convert value from utf8 to latin1
```

✓ Validation

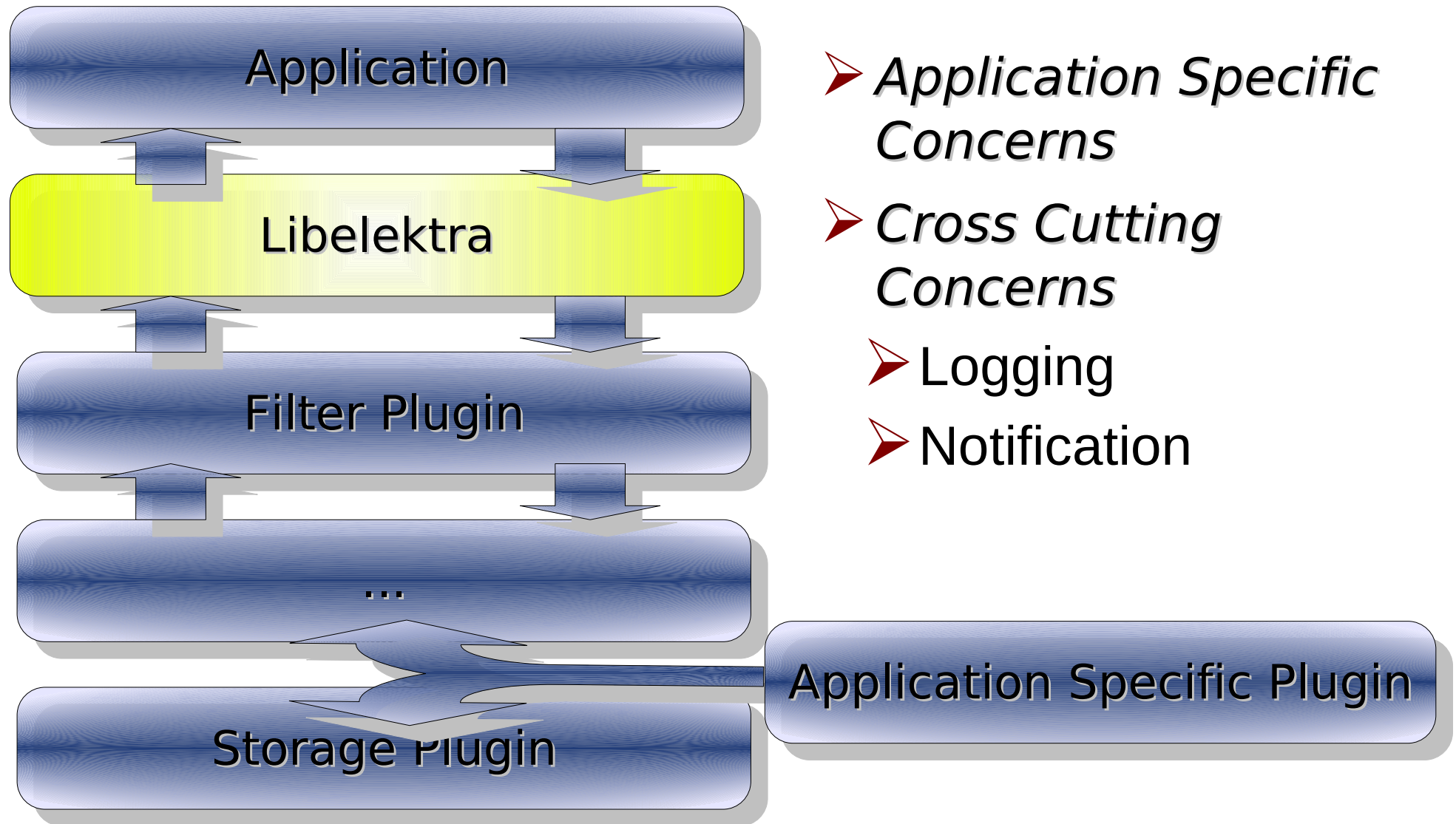
```
# kdb set system/sw/xorg/Screen/Display/Modes 12y80x1024  
Could not validate "12y80x1024", needed [0-9]+x[0-9]+
```

✓ Type System

```
# kdb set system/sw/xorg/Mouse/Option/Emulate3Buttons x  
Could not set "x", type boolean required.
```



Anwendung von Plugins (2)

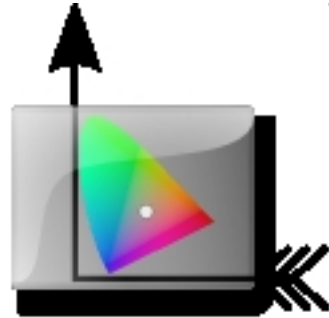


Einsatz von Elektra

✓ *Embedded Systems*



✓ *Cross Plattform Software*



ουρανός

Oyranos

Resources

- Homepage: <http://www.libelektra.org>
- Bugs: <http://bugs.libelektra.org>
- Svn: <http://svn.libelektra.org>
- API: <http://www.libelektra.org/elektra-api/>
- elektra@markus-raab.org
- Mailing Liste: registry-list@sf.net

<http://www.libelektra.org>





Ende

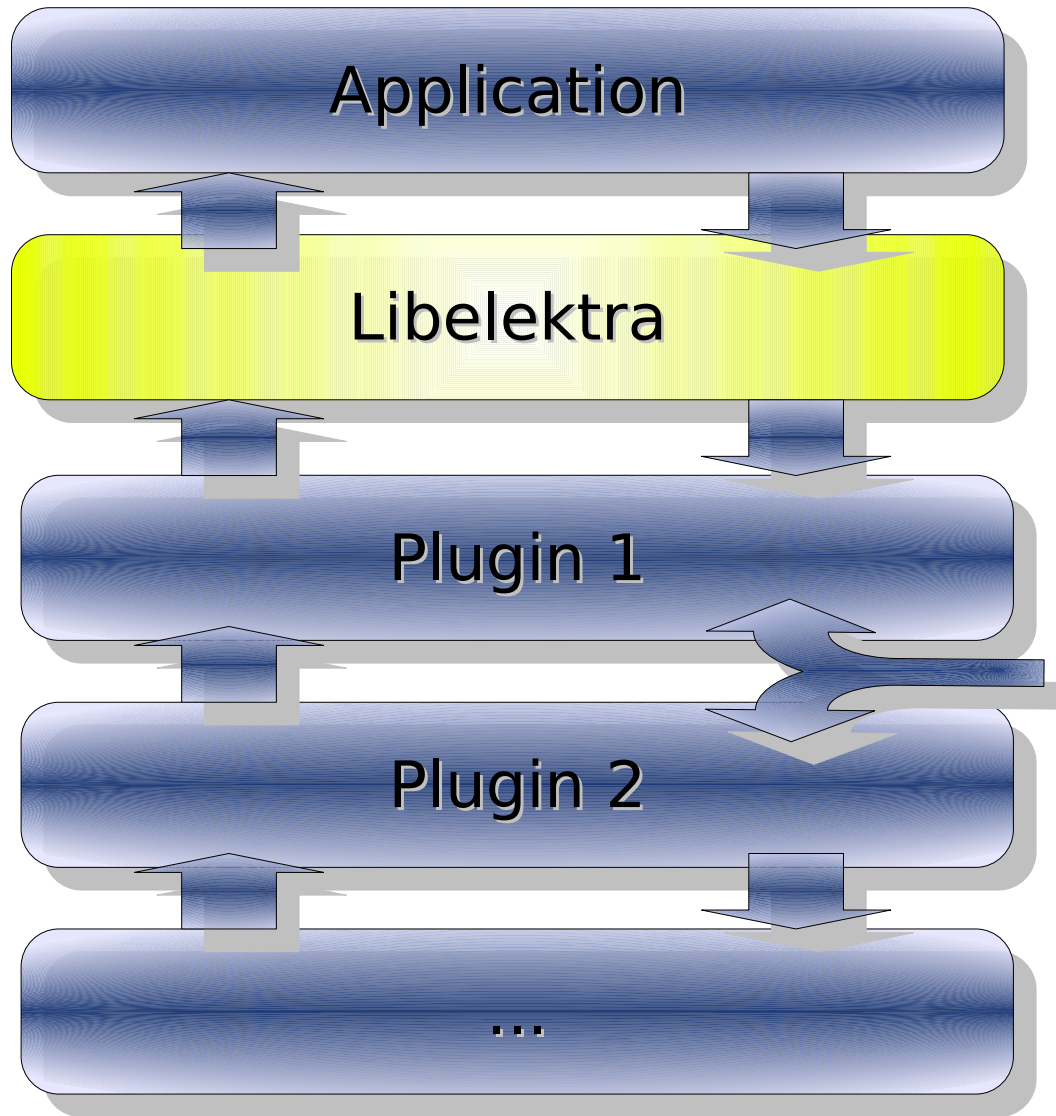
*Vielen Dank für Deine
Aufmerksamkeit!*

Markus Raab <elektra@markus-raab.org>

<http://www.libelektra.org>



Offene Fragen (1)



- *Abhängigkeiten zwischen Plugins?*
- *Verträge zwischen Plugins?*
- *Nicht selbst in den Fuß schießen*



Quelle: haldimandcountyhydro.ca

Offene Fragen (2)

```
LABEL=stable / jfs defaults,errors=remount-ro 0 1
proc /proc proc defaults 0 0
LABEL=swap none swap sw 0 0
LABEL=home /home jfs rw,suid,dev,exec,auto,nouser 0 2
```

✗ z.B. /etc/fstab akzeptiert nicht beliebiges

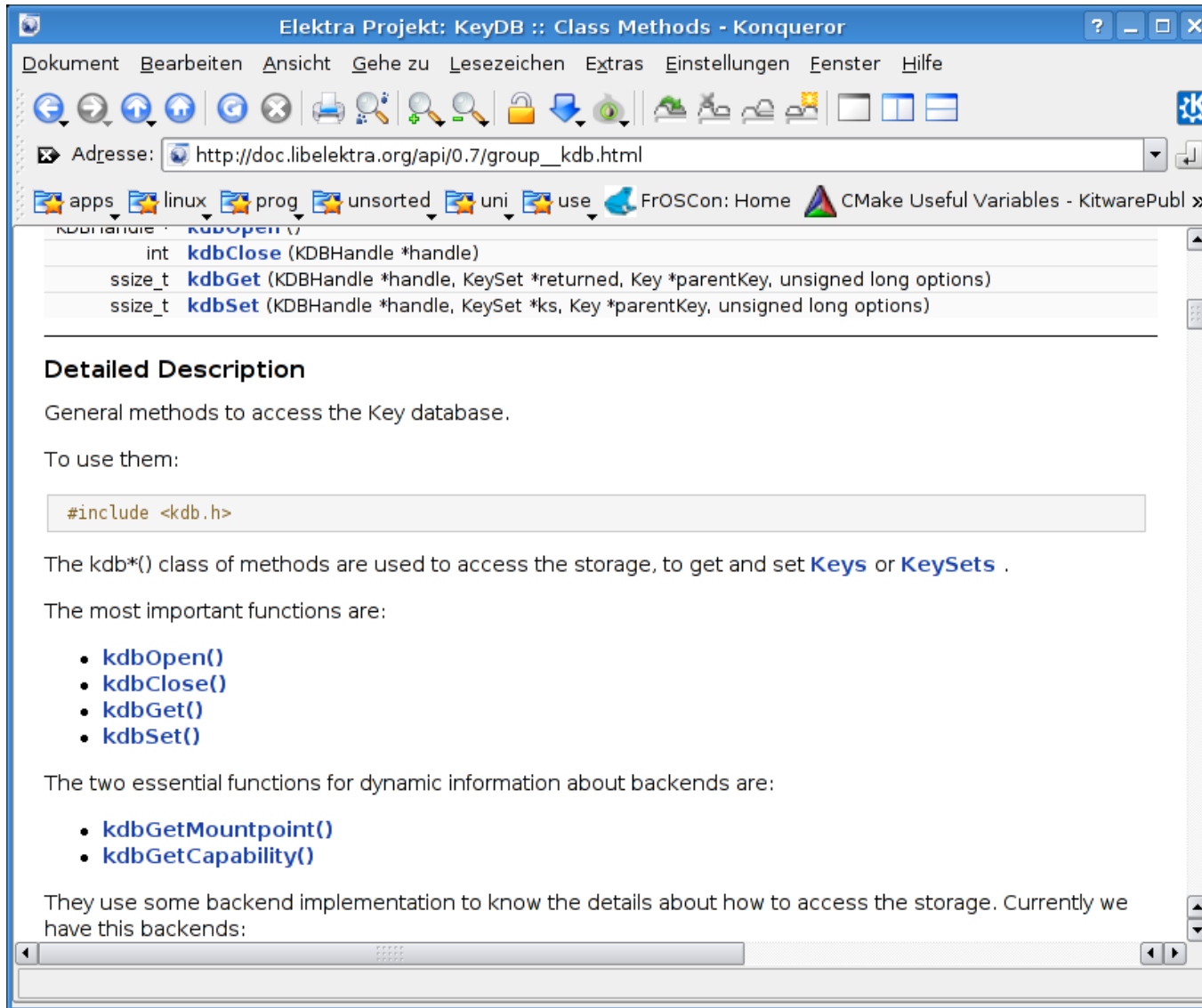
✗ Bestimmte Einträge müssen vorhanden sein

```
enum fstype
{fat,ntfs,ext2,jfs,proc,swap};
struct fstabEntry {
    string device;
    path mpoint;
    fstype type;
    string options;
    unsigned short dumptfreq;
    unsigned short passno;
};
```

- *Wie kann man obigen Typ beschreiben?*
- *Können Plugins dadurch beschreiben was sie akzeptieren?*

The API is Fully Documented!

<http://doc.libelektra.org/api/0.7/>



The screenshot shows a web browser window titled "Elektra Projekt: KeyDB :: Class Methods - Konqueror". The address bar displays "http://doc.libelektra.org/api/0.7/group__kdb.html". The main content area shows the API documentation for the `kdb` class methods. It lists three methods: `kdbClose`, `kdbGet`, and `kdbSet`. Below the list, there is a "Detailed Description" section with the text "General methods to access the Key database." and "To use them:" followed by a code block containing `#include <kdb.h>`. The description continues with "The kdb*() class of methods are used to access the storage, to get and set **Keys** or **KeySets** ." and "The most important functions are:" followed by a bulleted list of `kdbOpen()`, `kdbClose()`, `kdbGet()`, and `kdbSet()`. It also mentions "The two essential functions for dynamic information about backends are:" followed by a bulleted list of `kdbGetMountpoint()` and `kdbGetCapability()`. The text concludes with "They use some backend implementation to know the details about how to access the storage. Currently we have this backends:".

- ✓ API doc in Doxygen
- ✓ API man pages
- ✓ Viele Code Beispiele
- ✓ Programing Tutorial

<http://www.libelektra.org>



Qualität

- *Genaue Doku!*
- *Testfälle*
- *API: <http://www.libelektra.org/elektra-api/>*
- *Testing Framework*
 - *2769 Zusicherungen*
- *Bug Database*
- *Stable Policy*



Related Work

- *Elektra with Mounting*
- *Uniconf (with Daemon)*
- *Debconf (supports stacking)*
- *Augeas (Linux Configuration Files only)*
- *Pam (Error Conditions)*
- *Puppet*

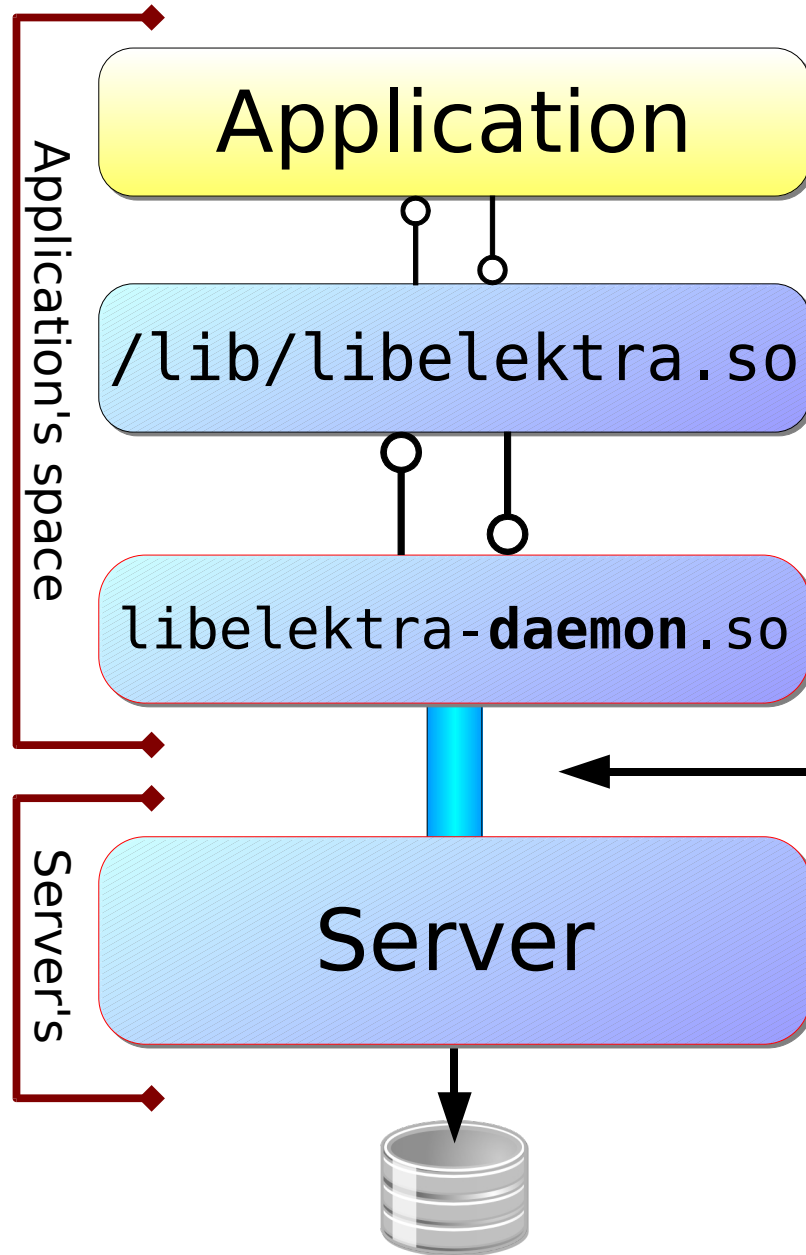


State-of-the-Art

- *M. Lackner, A. Krall, and F. Puntigam. Supporting design by contract in Java. Journal of Object Technology, 1(3):57–76, 2002.*
- *C. Andreae, J. Noble, S. Markstrum, and T. Millstein. A framework for implementing pluggable type systems. ACM SIGPLAN Notices, 41(10):74, 2006.*
- *Markus Raab and Patrick Sabin. Implementation of Multiple Key Databases for Shared Configuration. <ftp://www.markus-raab.org/elektra.pdf>, March 2008.*



Dämon Architektur



→ Client-Server Architecture

- ◆ Kommunikation zum Server über [libelektra-daemon.so](http://www.libelektra.org/libelektra-daemon.so)
- ◆ Der Server selber verwendet auch [libelektra.so](http://www.libelektra.org/libelektra.so) um key/value zu speichern.
- ◆ single point of failure

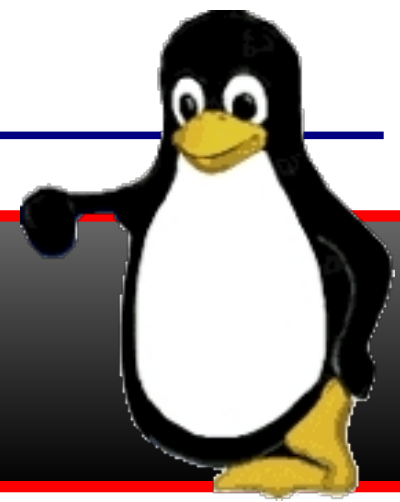
● Protokol über unix domain sockets

GConf ist fett und dependent...

```
bash$ ldd /usr/lib/libgconf-2.so.4
libgobject-2.0.so.0 => /usr/lib/libgobject-2.0.so.0
libORBit-2.so.0 => /usr/lib/libORBit-2.so.0
libm.so.6 => /lib/tls/libm.so.6
libgmodule-2.0.so.0 => /usr/lib/libgmodule-2.0.so.0
libdl.so.2 => /lib/libdl.so.2
libgthread-2.0.so.0 => /usr/lib/libgthread-2.0.so.0
libglib-2.0.so.0 => /usr/lib/libglib-2.0.so.0
libpthread.so.0 => /lib/tls/libpthread.so.0
libc.so.6 => /lib/tls/libc.so.6
libpopt.so.0 => /usr/lib/libpopt.so.0
/lib/ld-linux.so.2
```



Elektra ist leicht . . .



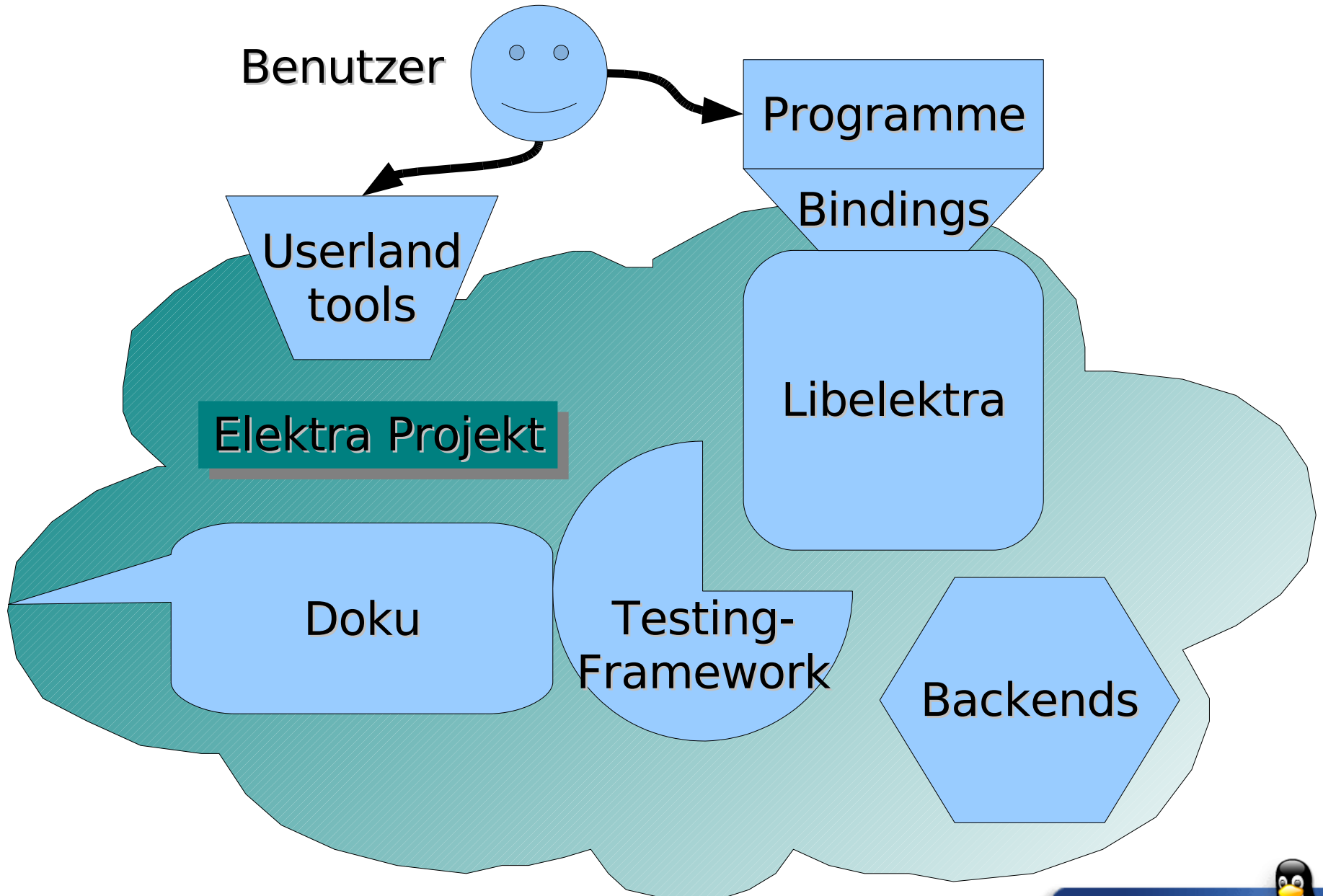
```
bash$ ldd /lib/libelektra.so
libc.so.6 => /lib/tls/libc.so.6
/lib/ld-linux.so.2 => /lib/ld-linux.so.2
```

- ✓ *Verwendbar bei Restriktionen (kein Netzwerk, kein /usr), sogar bei /sbin/init*
- ✓ *Auch ohne Dämon verwendbar*
 - ◆ *No single point of failure*
 - ◆ *Kein Kommunikationsprotokoll notwendig*
 - ◆ *Keine Sicherheitslöcher möglich*
 - ◆ *Sicherheit über OS*
 - ◆ *Einfach und sauber*

<http://www.libelektra.org>



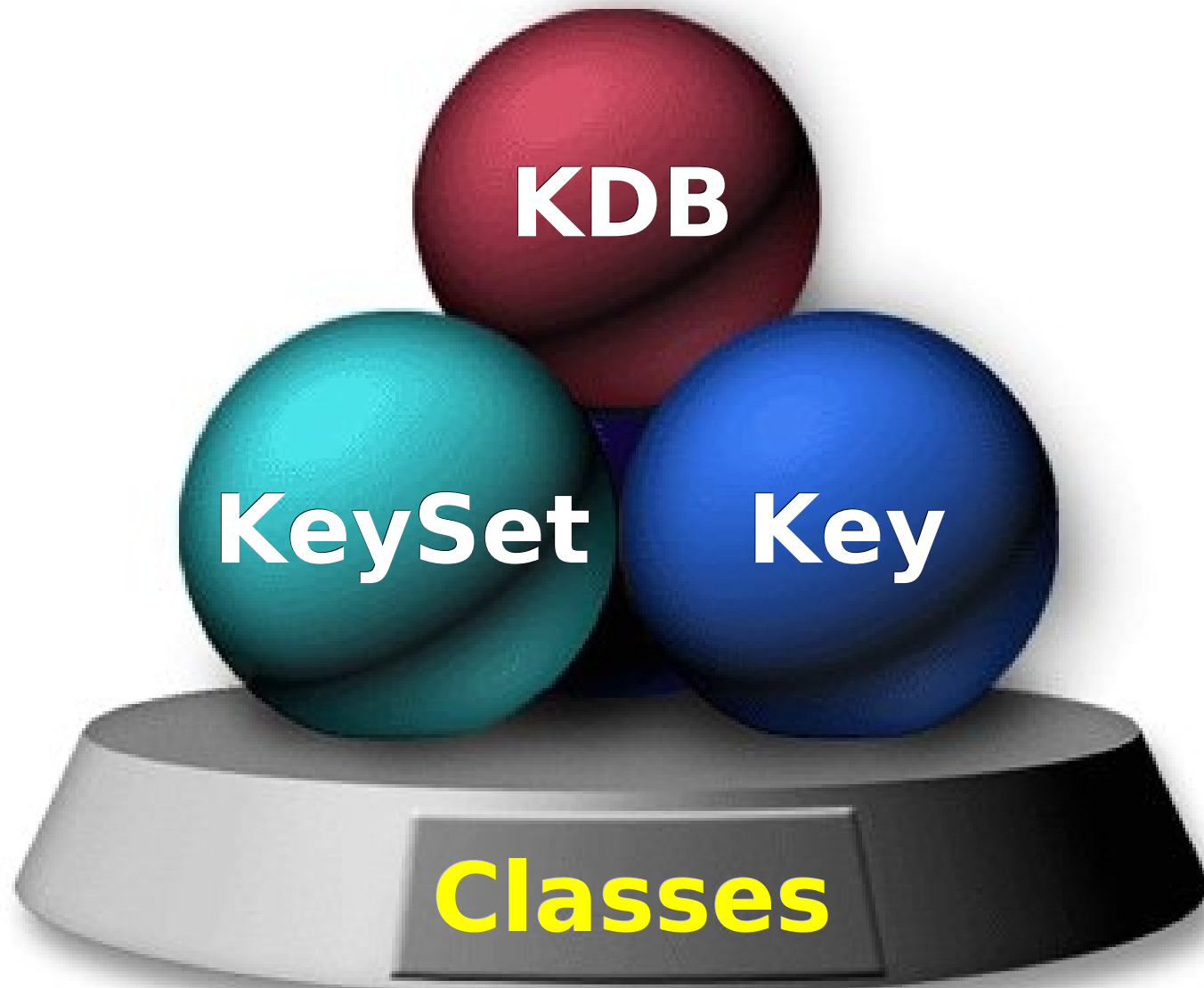
Ecosystem



Die API

*Was ist vorhanden um auf Elektra
key/value pairs zuzugreifen?*

Nur 3 Klassen



Key Eigenschaften

system/users/root/shadowPassword = \$1\$yM93nU

user:valeria/env/env2/PATH = \$PATH:/usr/sbin



- *Name*
- *Value*
- *User domain or owner*
- *Kommentar*
- *System UID & GID*
- *Access Permissions*
- *Last change time*
- *Last access time*

Keyset Eigenschaften



- *Container für Keys*
- *Menge*
- *Effiziente Übergabe vieler Keys*
- *Schnelles Suchen nach Namen*

Key Data Base



- *Abstrahiert Zugriff auf permanente Keys*
- *Liest oder schreibt Keysets auf die Festplatte*

Fertig um jetzt benutzt zu werden.

➤ Vorhanden

- ✓ C
- ✓ C++
- ✓ Shell
- ✓ XML



➤ War mal da . . .

- ✓ Python
- ✓ Scheme
- ✓ Ruby
- ✓ Java

C API Methods

Class KDB

```
kdbOpen()  
kdbClose()  
kdbGet()  
kdbSet()  
kdbGetKey()  
kdbSetKey()  
kdbGetString()  
kdbSetString()  
kdbRemove()
```

Class Key

```
keyNew() keyGetUID()  
keyDup() keySetUID()  
keyCopy() keyGetGID()  
keyClear() keySetGID()  
keyDel() keyGetAccess()  
keyGetAccess()  
keySetAccess()  
keyGetOwner()  
keySetOwner()  
keyGetCommentSize()  
keyGetComment()  
keySetComment()  
keyGetValueSize()  
keyGetString()  
keyGetBinary()  
keySetBinary()  
keyValue()  
  
keyGetMTime()  
keyGetATime()  
keyGetCTime()
```

Class KeySet

```
ksNew()  
ksDel()  
ksInsertKey()  
ksInsert()  
ksAppendKey()  
ksAppend()  
  
ksNext()  
ksRewind()  
  
ksLookup()  
ksLookupByName()  
ksLookupByValue()
```

The Elektra Initiative needs YOU !

- *Avi Alkalay* <avi@unix.sh>
- *Markus Raab* <elektra@markus-raab.org>
- *Yannick Lecaille* <yl@itioweb.com>
- *Jens Andersen* <jens.andersen@gmail.com>
- *Patrick Sabin* <patrick@sabin@gmx.at>
- *Pier Luigi Fiorini* <pierluigi.fiorini@mockup.org>
- *Rèmi* <remipouak@yahoo.fr>
- *Studio-HB* <contact@studio-hb.com>

<http://www.libelektra.org>

