A Modular Approach to Configuration Storage

Linuxwochenende 2010

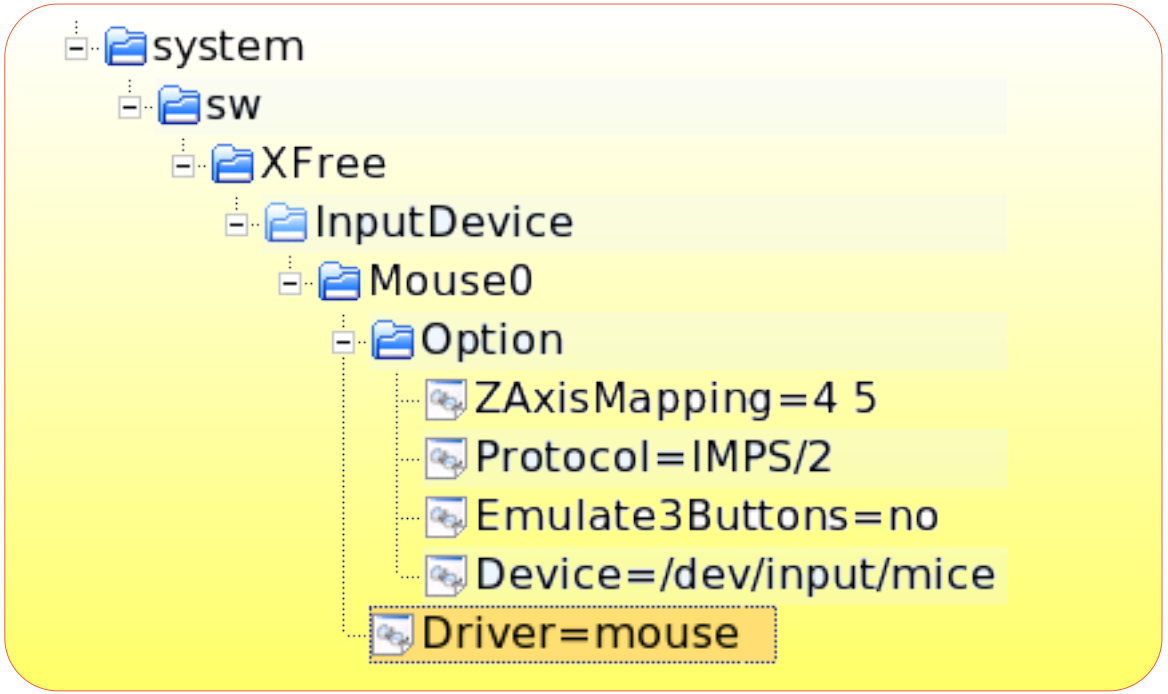
Metalab

Markus Raab <elektra@markus-raab.org>

Xmpp: jabber@markus-raab.org

http://www.libelektra.org

Was ist Konfiguration?

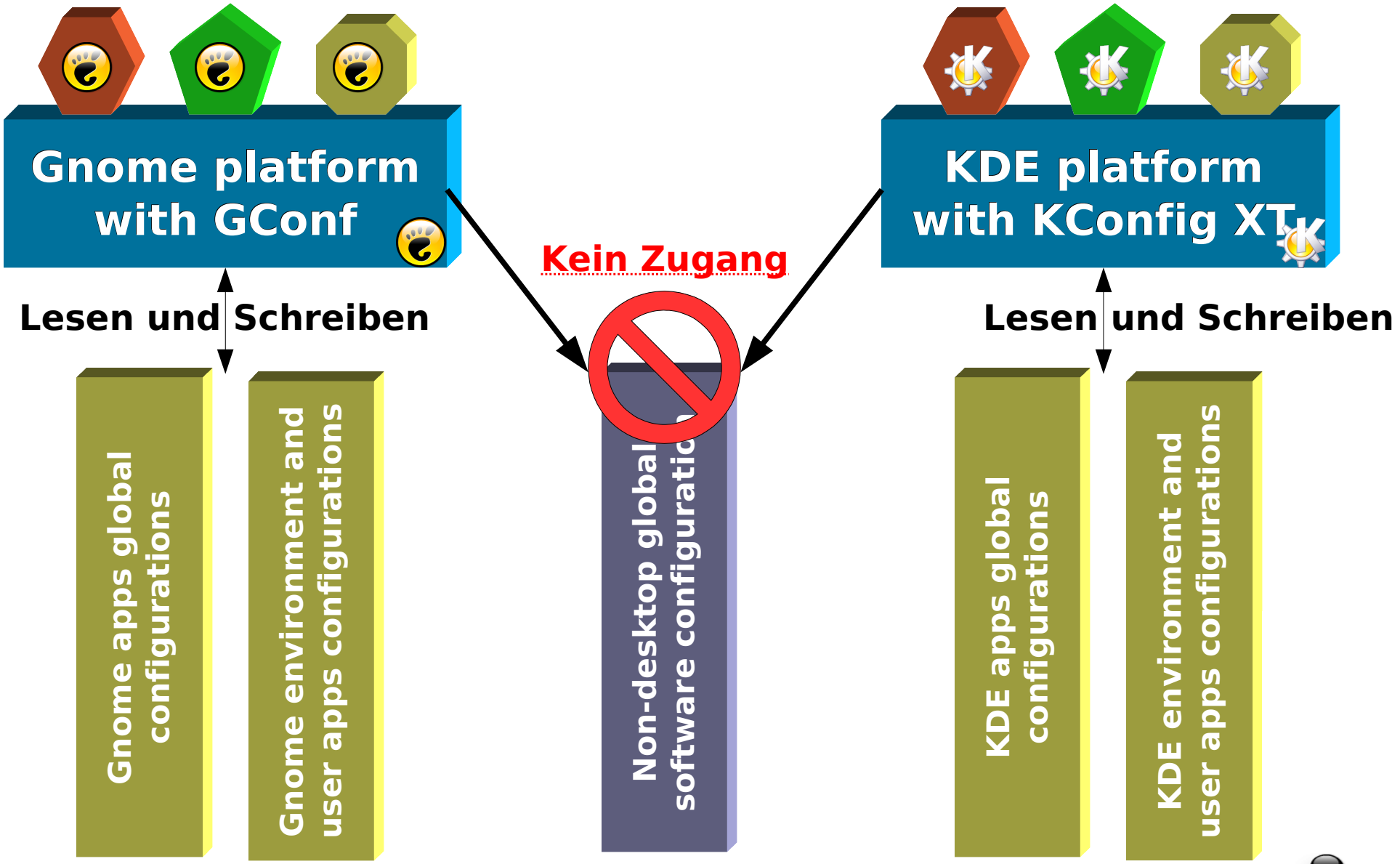


- ✔ Einstellungen und Optionen von Programmen
- ✔ Einfache hierarchische Struktur

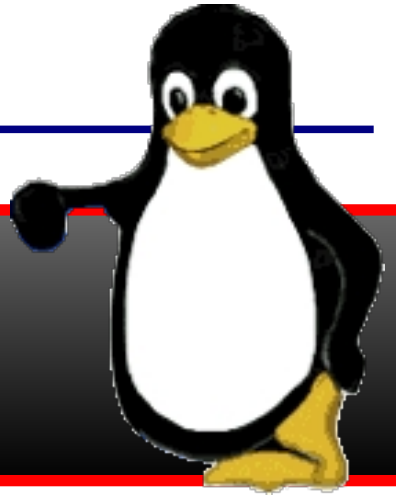
```
Section "InputDevice"
        Identifier  "Mouse0"
        Driver      "mouse"
        Option      "Protocol" "IMPS/2"
        Option      "Device" "/dev/input/mice"
        Option      "ZAxisMapping" "4 5"
        Option      "Emulate3Buttons" "no"
EndSection
```

- ✔ Für Menschen lesbar
- ✔ Für Menschen editierbar
- ✗ Nicht für alle Programme einfach lesbar/editierbar

Konfiguration für jede Platform?



Was ist Elektra?



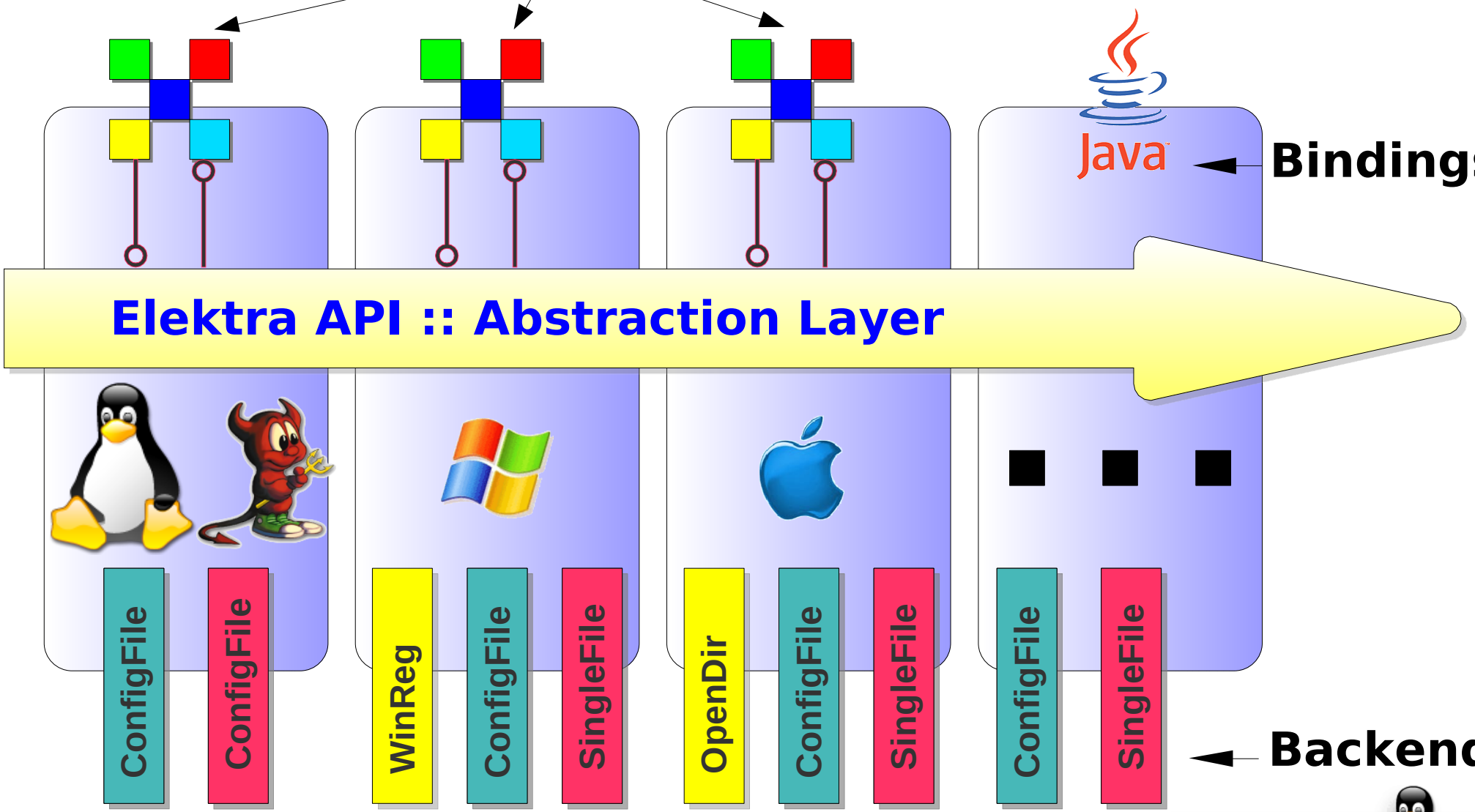
```
sh$ ldd /lib/libelektra.so
    libc.so.6 => /lib/tls/libc.so.6
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2
```

- ✔ *Bibliothek um Konfiguration zu speichern*
- ✔ *Klein und portabel*
- ✔ *Während Diplomarbeit erweitert*
- ✔ *Sicherheit über OS*
- ✔ *Konfigurationsdateien*
- ✔ *Abstraktion von Konfiguration*

Was ist das Ziel?

Portable und nicht portable Software

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

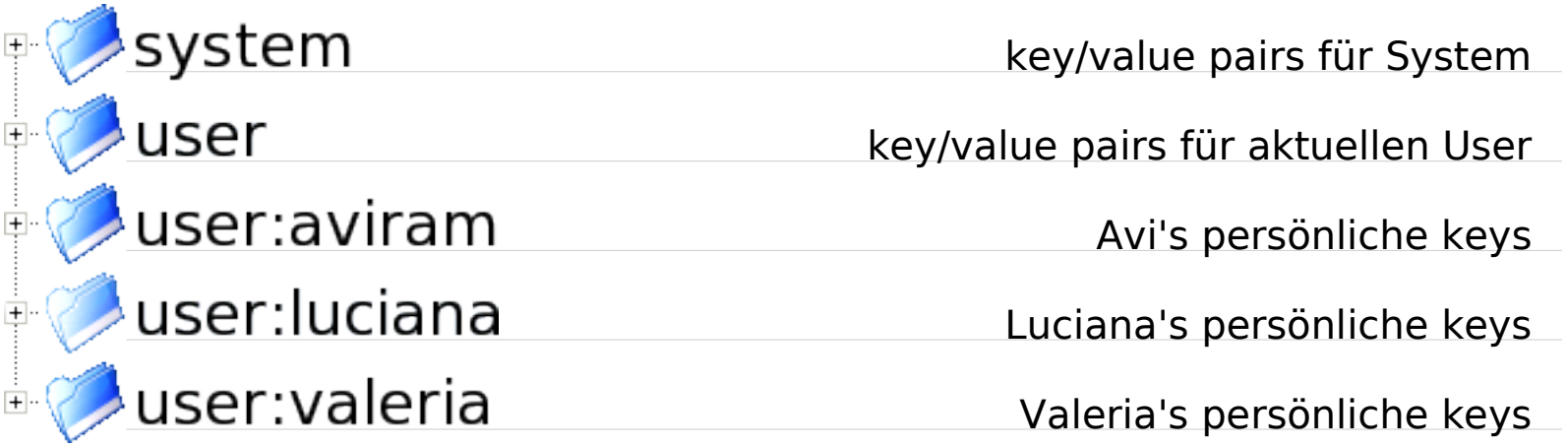


Key Database Semantik

Linuxwochenende Wien 2010

Markus Raab <elektra@markus-raab.org>

Key Hierachie

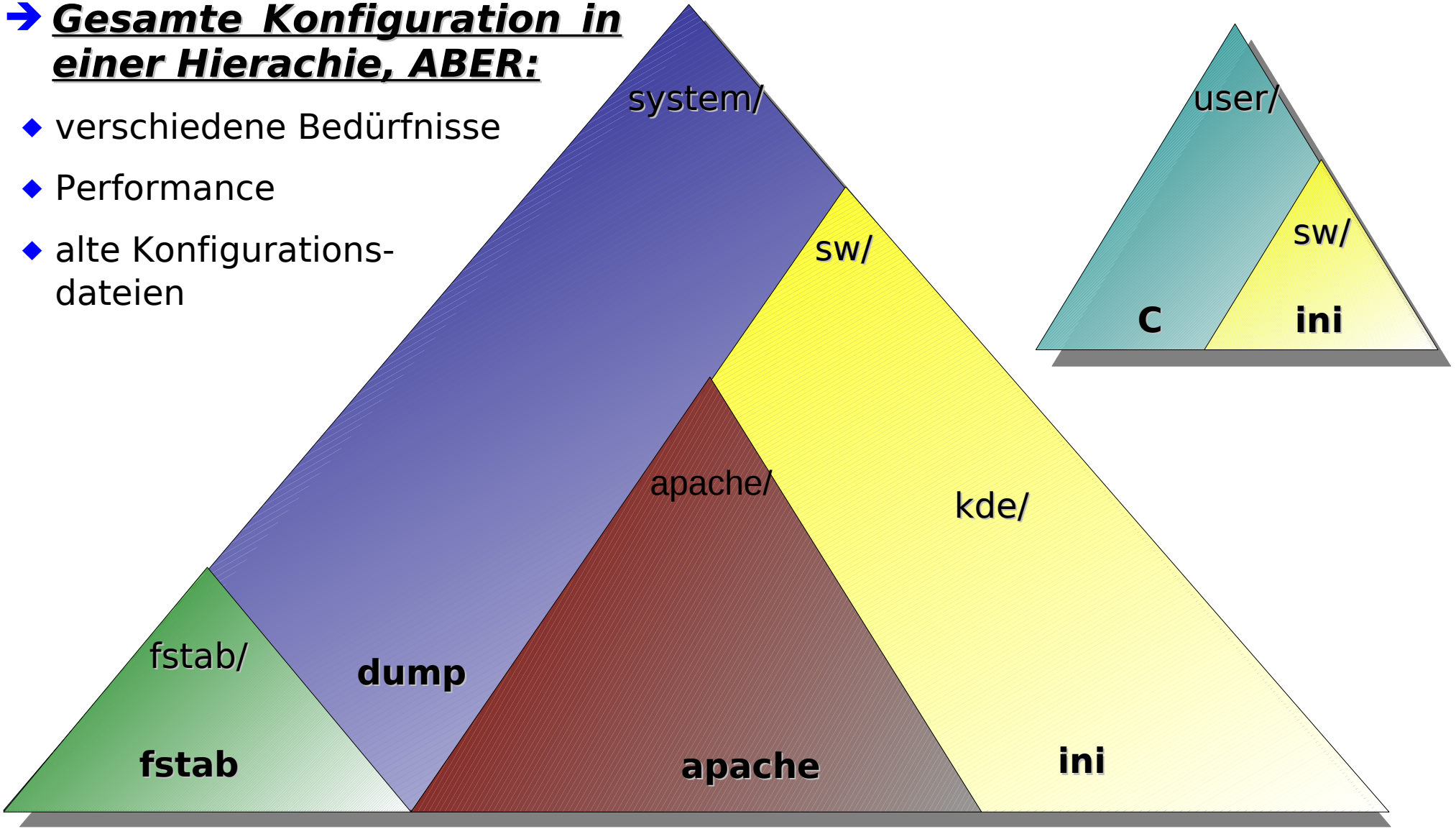


- Die ***system/**** Hierachie ist gespeichert unter ***/etc/***
- Die ***user:$USER/**** Hierachie unter ***~$USER/.config/***
- Die ***user/**** Hierachie ist eine Abkürzung für aktuellen User.

Key Hierachie :: *system/...*

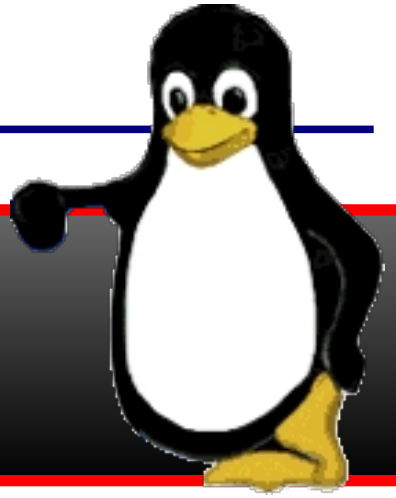
Backends mounten

- ***Gesamte Konfiguration in einer Hierachie, ABER:***
  - verschiedene Bedürfnisse
  - Performance
  - alte Konfigurations-dateien



`system/sw/apache/site1.com/DocRoot`

Bootstrapping



```
sh$ kdb mount
    .. load default backend
    .. buildup other backends
```

- ✔ *Self-Contained Configuration System*
- ✔ *eingebautes Default Backend zuerst*
- ✔ *Damit eigene Konfiguration einlesen*
- ✔ *Backends in globalen Konfigurationsbaum mounten*

KDB Semantik

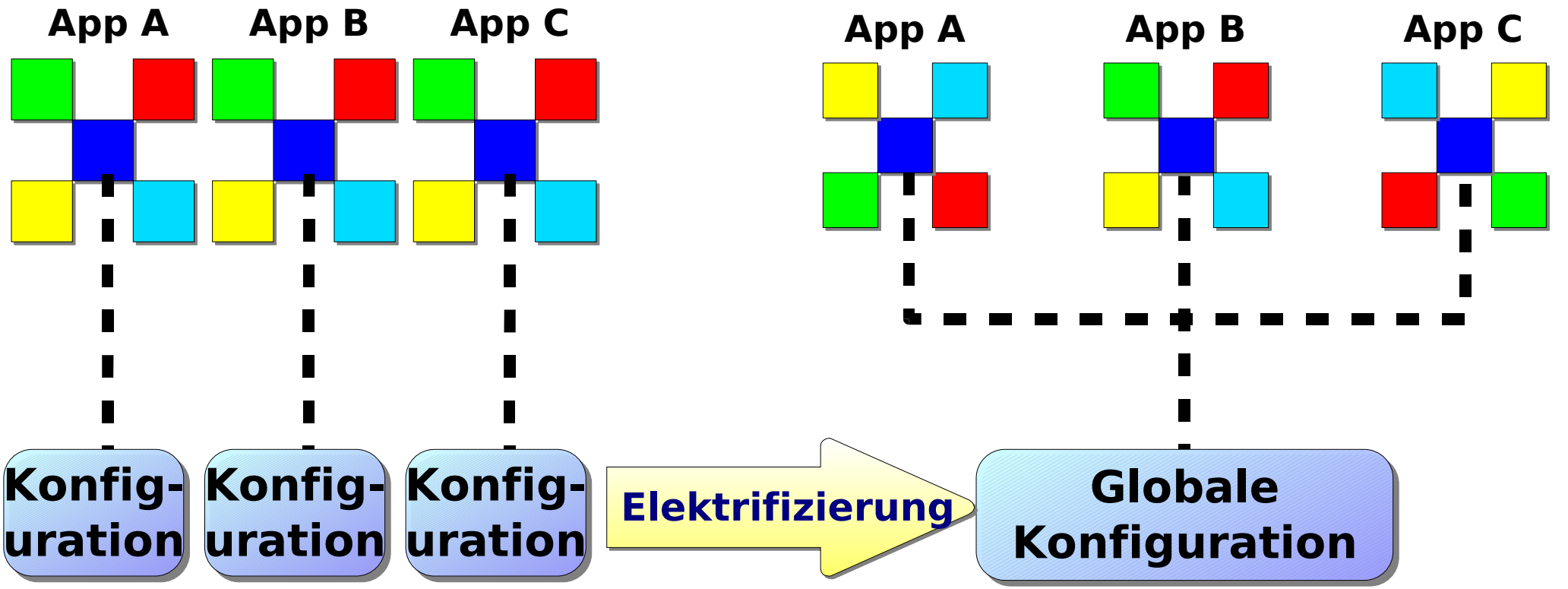
1. Wurzel: User und System
2. Keys sind mit Namen **eindeutig** definiert
3. *Löcher* werden akzeptiert
4. Keine Ordnungsrelation (Set)
5. Keys sind unabhängig
6. Keine Links, Verstecke Keys o.ä. FS-Semantiken
7. Aber Plugins und Metadaten können 3-6 abändern

Elektrifizierung

Linuxwochenende Wien 2010

Markus Raab <elektra@markus-raab.org>

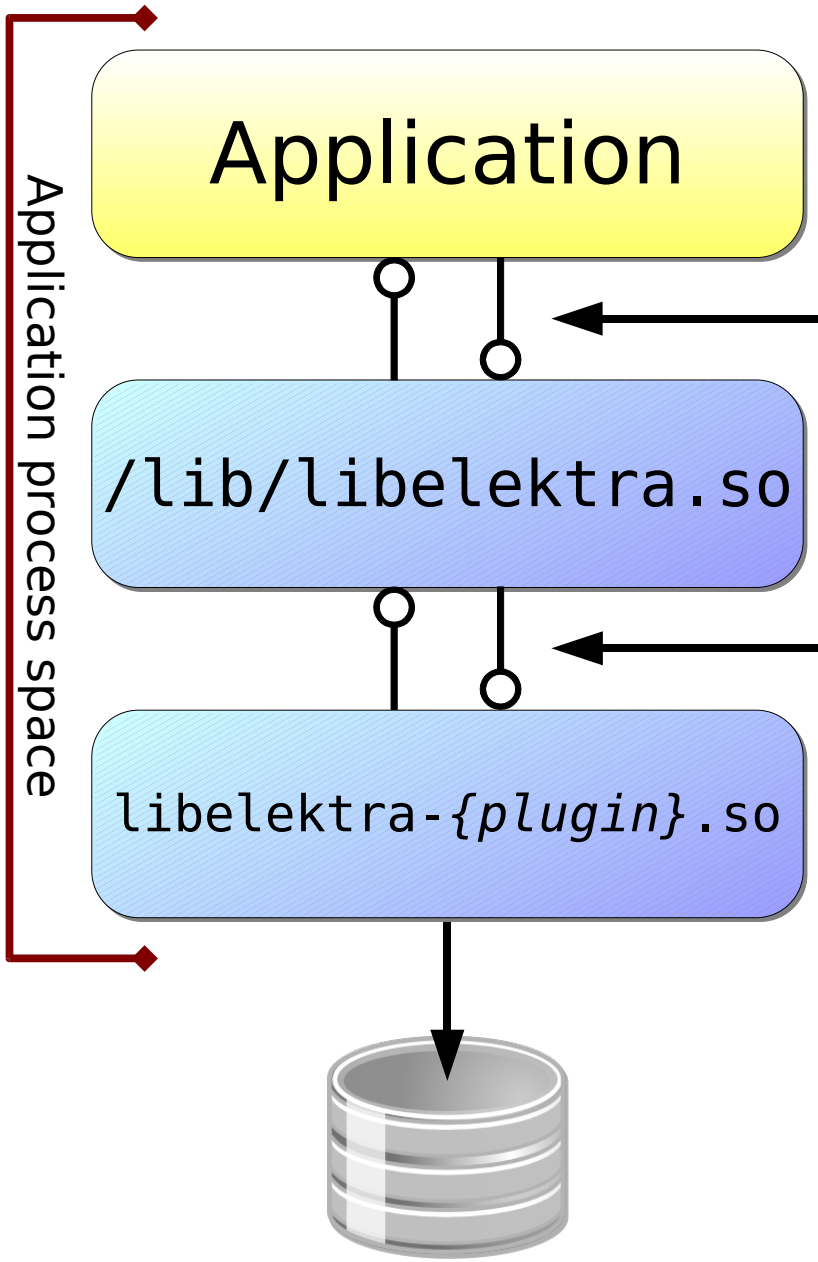
Elektrifizierung



- ✗ 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

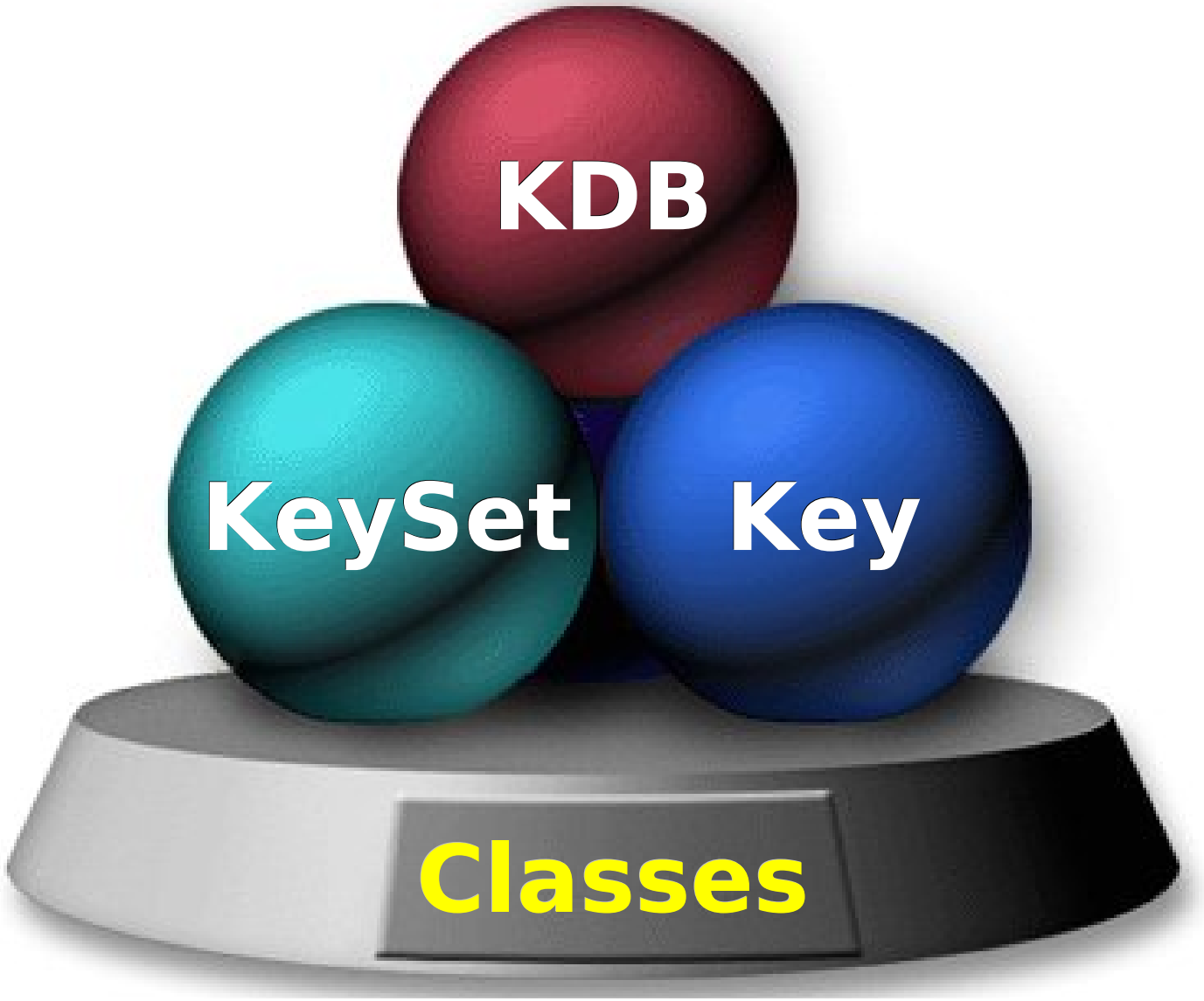


- ***Dynamic single dependency***
  - ✔ Sichtbar mit *ldd*
  - ✔ *libelektra.so* versteckt Backends
- ***Pure dynamic (in)dependency***
  - ✔ Plugin = Shared Library (Linux)
  - ✔ Nicht sichbar mit *ldd*
  - ✔ Je nach Pfad variabel (Mounting)
- ***Different storages***

Die API

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

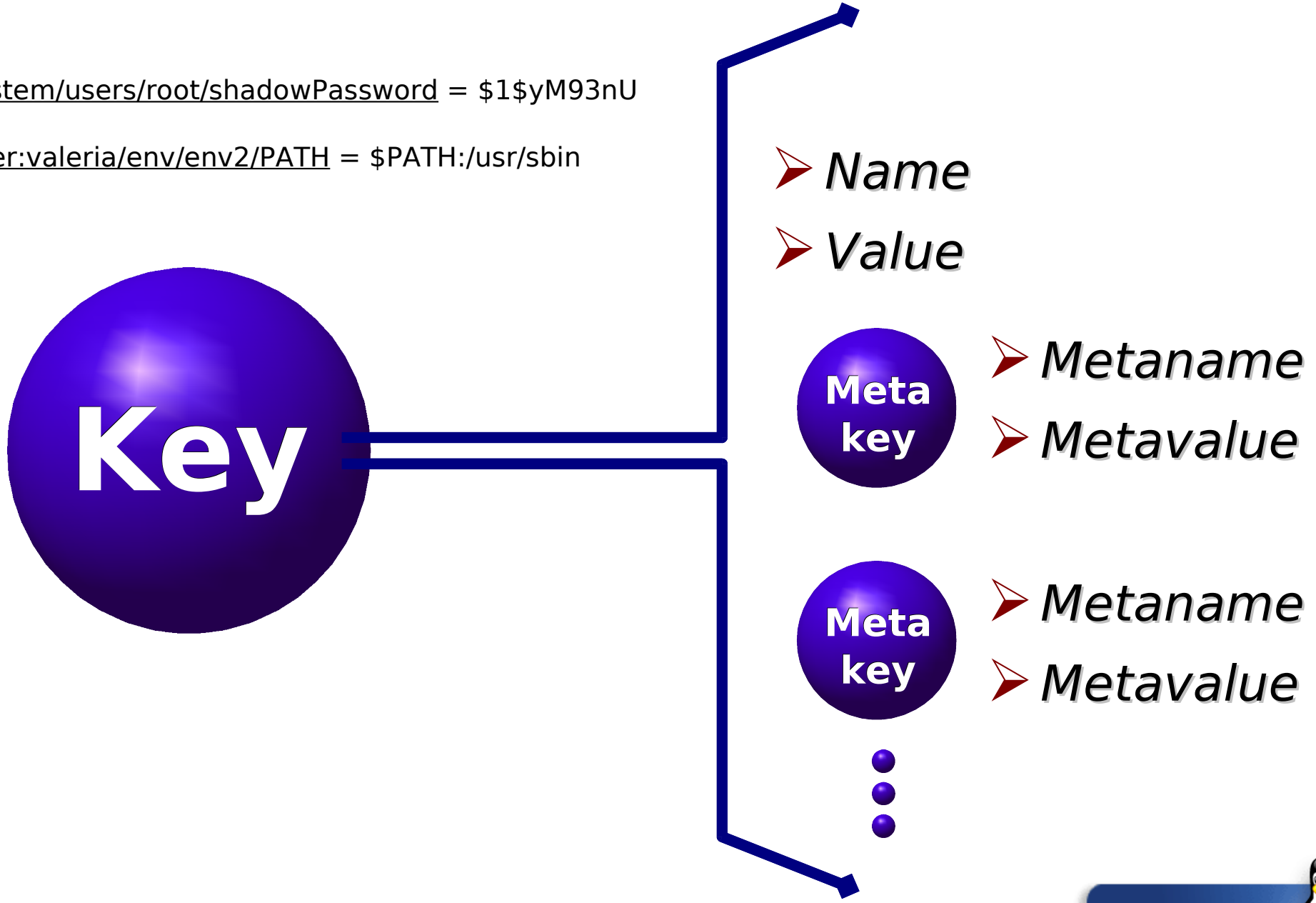
Nur 3 Klassen



Key Eigenschaften

system/users/root/shadowPassword = 1yM93nU

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



C API Methods

Class KDB

kdbOpen()
kdbClose()
kdbGet()
kdbSet()

Class Key

keyNew()
keyDup()
keyCopy()
keyClear()
keyDel()

keyIncRef()
keyDecRef()
keyGetRef()

keyRewindMeta()
keyNextMeta()
keyCurrentMeta()

keyGetMeta()
keySetMeta()

keyGetString()
keyGetBinary()
keySetBinary()
keyValue()

Class KeySet

ksNew()
ksDel()
ksAppendKey()
ksAppend()

ksNext()
ksRewind()

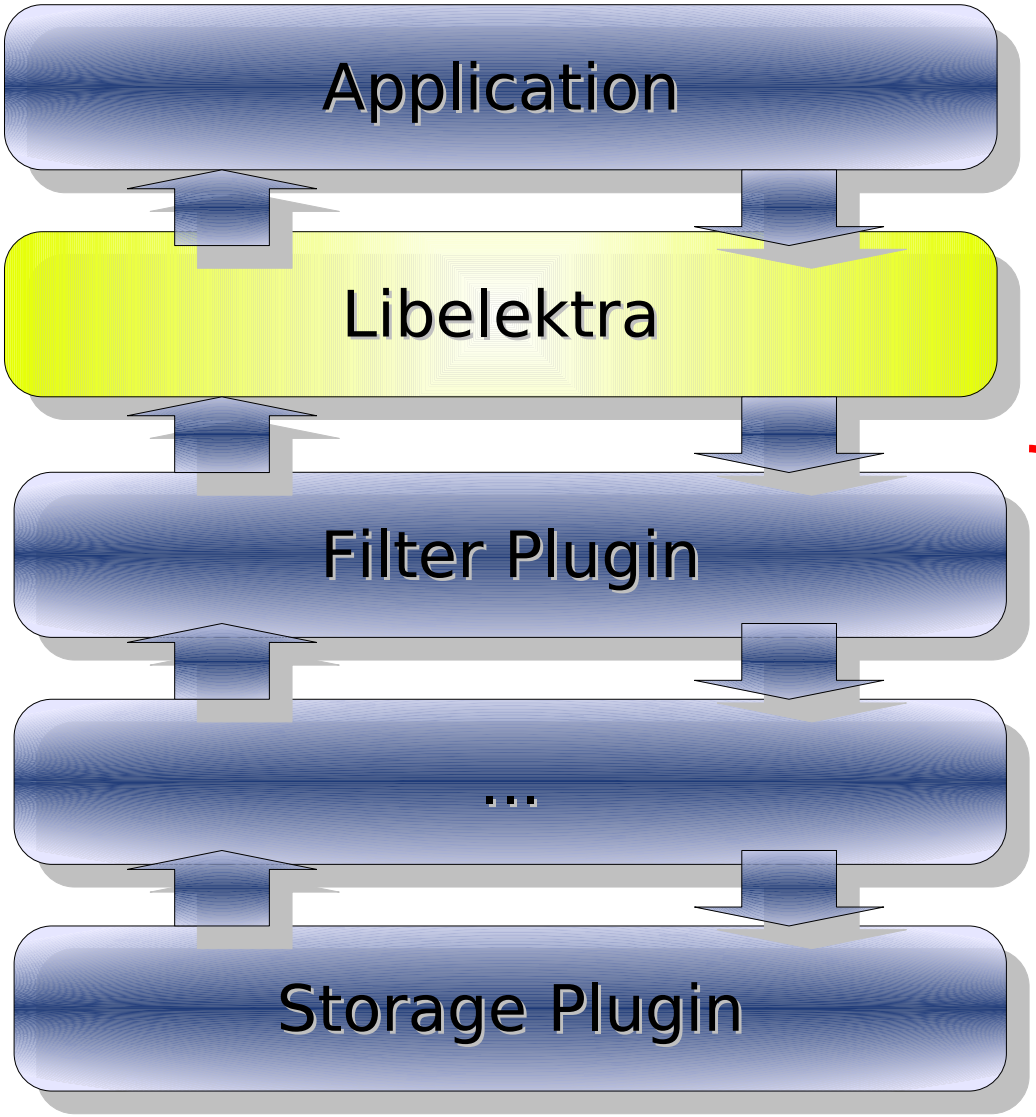
ksLookup()
ksLookupByName()

Modularer Ansatz

Linuxwochende 2010

Markus Raab <elektra@markus-raab.org>

Erwünschte Situation



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

n Plugins = 1 Backend

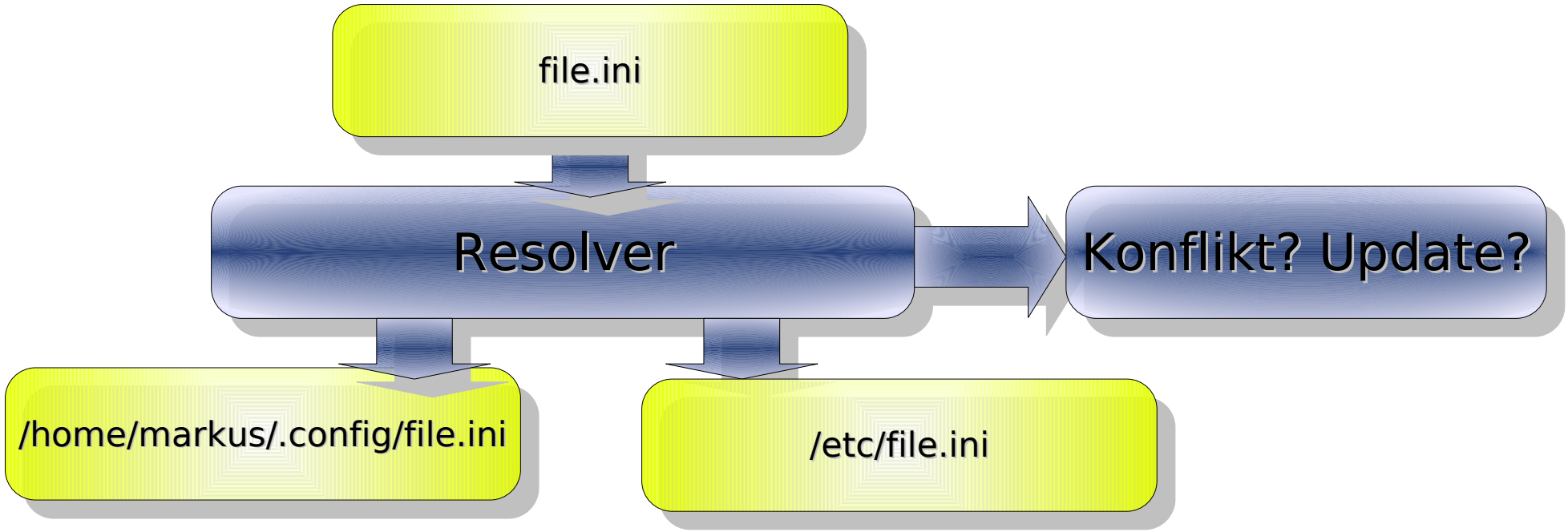
Plugins

Linuxwochende 2010

Markus Raab <elektra@markus-raab.org>

Resolver Plugin

- *Erledigt Betriebsystemabhängiges*
  - *Ermittelt Dateinamen*
  - *Erkennt Konflikte*
  - *Erkennt ob Update notwendig*



Storage Plugins

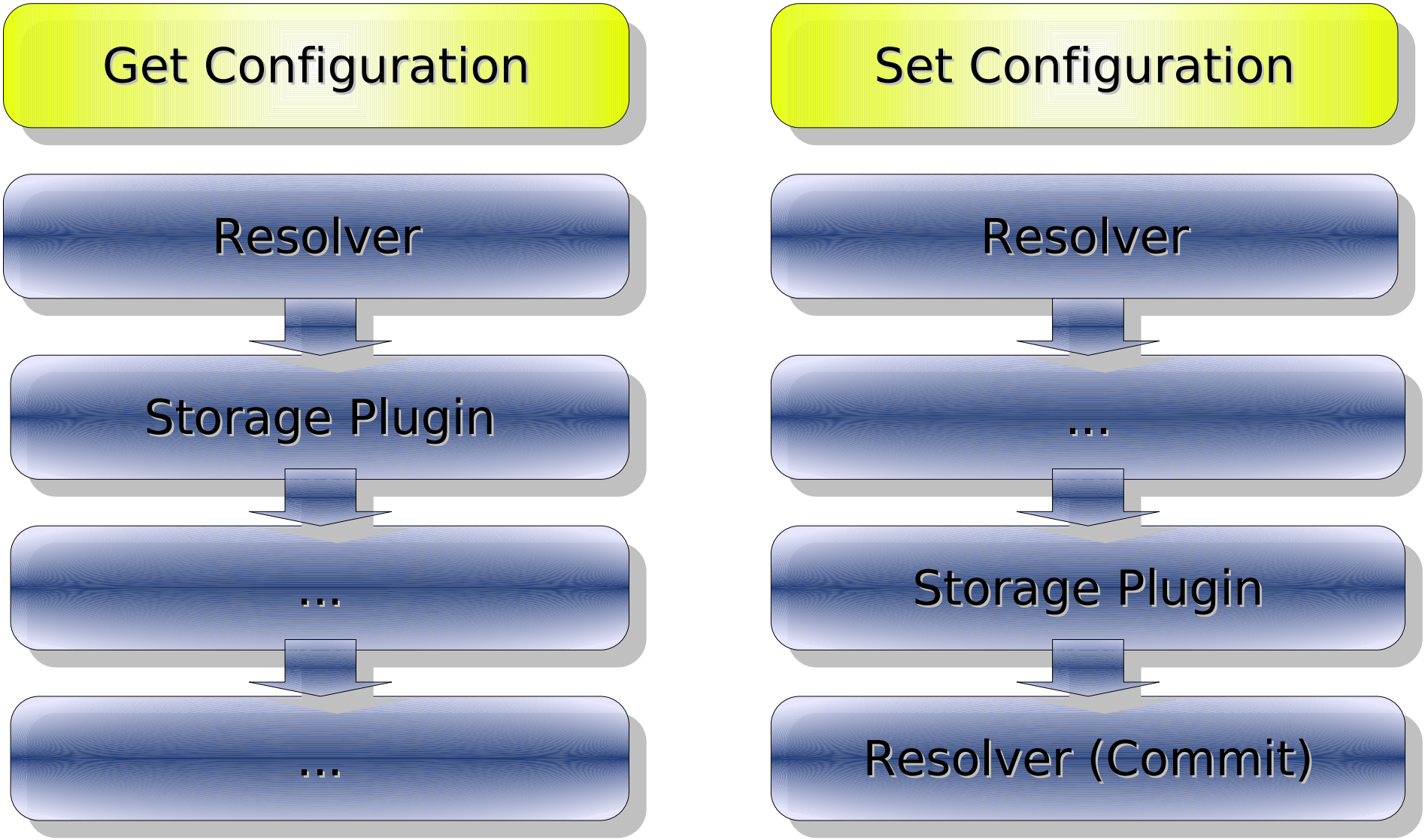
- *Dump*

- *XML*

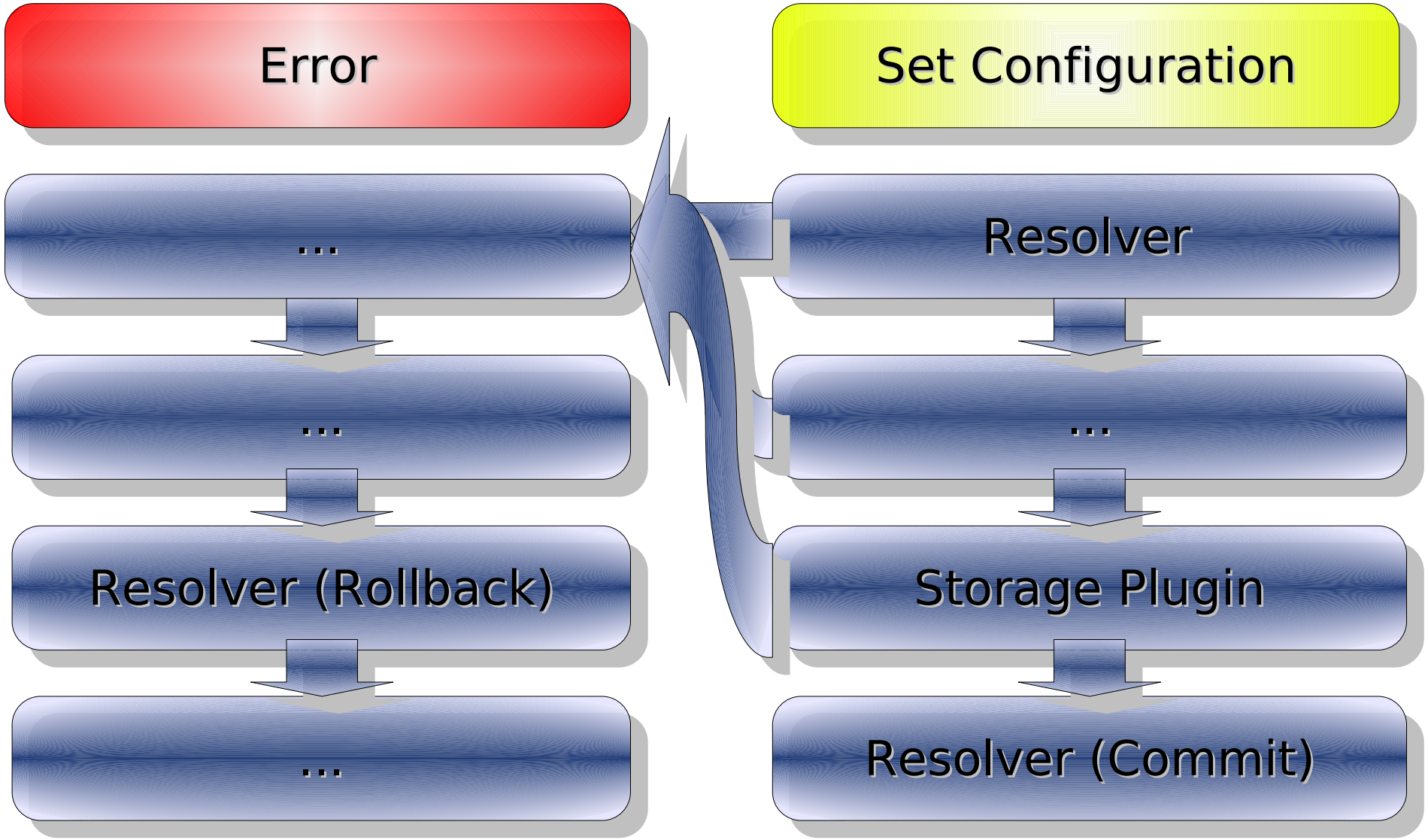
```
kdbOpen 1
ksNew 3
        keyNew 10 8
        user/dumprootkey
        keyEnd

        keyNew 15 7
        user/dump/key1value1
        keyEnd

        keyNew 15 7
        user/dump/key2value2
        keyEnd


ksEnd
```

```
<?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>
```

Reihenfolge von Plugins



Fehlerbehandlung



The *kdb* Command: Perfekt für Skripte

```
$ kdb get system/filesystems/boot/mpoint

# kdb set system/sw/xorg/Screen/Display/Modes 1280x1024


$ kdb ls system/sw/myapp

$ kdb rm user/env/alias/vnc
```

Cross-Cutting-Concerns

- ✔ *Logging (Syslog)*

```
Aug 31 10:25:05 markusbyte elektra[4149]:
  committed configuration user/syslog with 1 keys
Aug 31 10:25:14 markusbyte elektra[4150]:
  committed configuration user/syslog with 2 keys
```

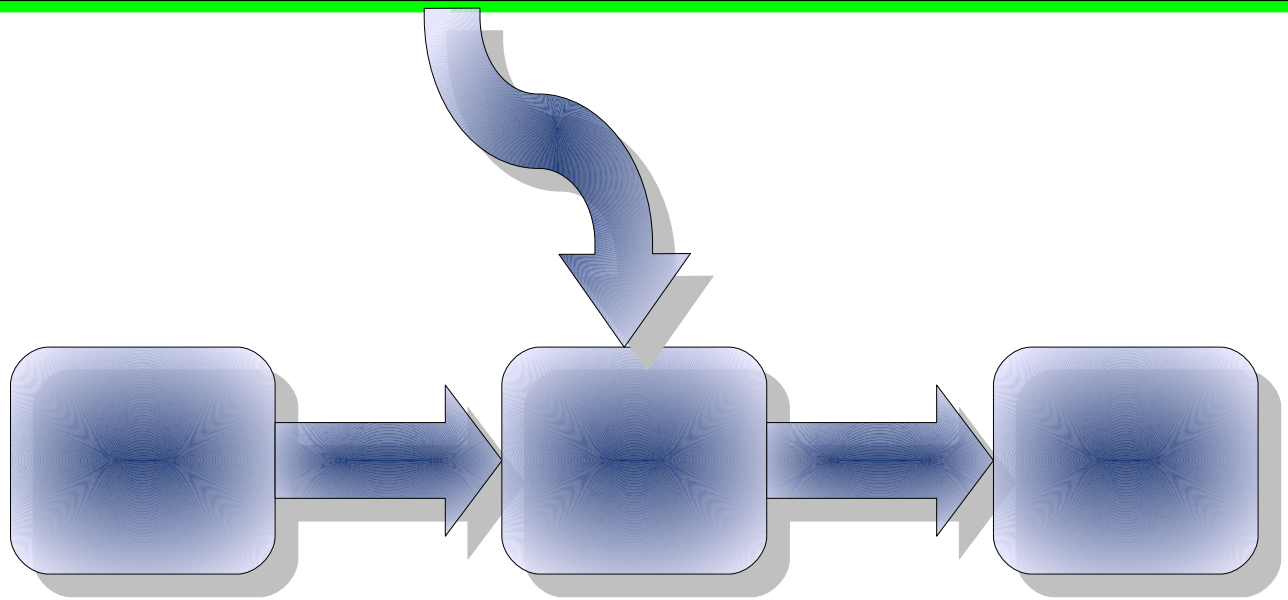
- ✔ Notification (D-Bus)

```
signal sender=:1.9 -> path=/org/libelektra/configuration;
                      interface=org.libelektra; member=changed
```

Metadaten

- ✔ *Daten über Konfiguration*
- ✔ *Globbing*

```
keyNew ("system/elektra/modules/hosts/config/needs/glob/#1",
        KEY_VALUE, "/*",
        KEY_META, "check/ipaddr", "",
        KEY_META, "validation/regex", "^[0-9.:]+$",
        KEY_END),
```



Basic Checker

- ✔ Validation

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

- ✔ *Network*

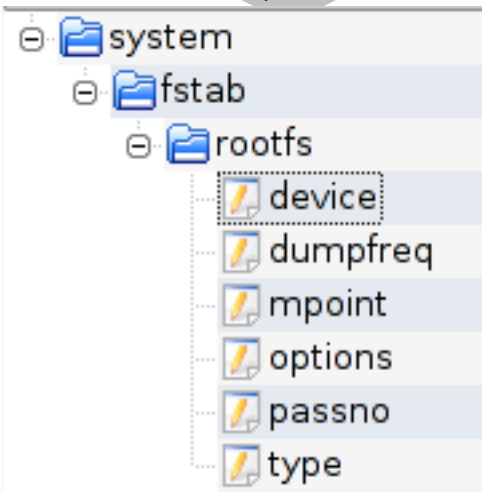
```
$ kdb set user/hosts/hostname 192.168.0.x
Value of key is not a valid IP Address
Name or service not known
```

- ✔ *Filename*

```
$ kdb set user/fstab/rootfs/device /dev/none
Warning: No such file or directory from key:
user/fstab/rootfs/device with path: /dev/none
```

Struct Checker

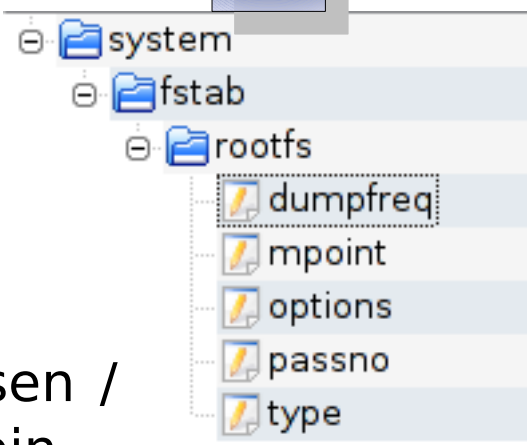
```
LABEL=stable /        jfs    defaults,errors=remount-ro 0 1
proc         /proc    proc   defaults 0 0
LABEL=swap   none     swap   sw 0 0
LABEL=home   /home    jfs    rw,suid,dev,exec,auto,nouser 0 2
```




```
struct FStab {
    string device;
    string mpoint;
    FSType type;
    string options;
    unsigned_short dumpfreq;
    unsigned_short passno;
};
```

Struct

list <FStab>



Änderungen

✗ Bestimmte Einträge müssen / dürfen nicht vorhanden sein

Type Checker

```
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
```

✗ /etc/fstab akzeptiert nicht beliebiges

```
enum FSType
{fat,ntfs,ext2,jfs,proc,swap};
string
unsigned_short
```

- *Überprüfung der Typen*
- *Basic CORBA Typsystem*

More Storage

- TCL

- Nickel

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

```
multi="line
inside quotes" ; with comments
after
escapes=\\\a\r\n\x11\b\056\t\\t\
xj
\==\; not a comment
\[not a section\]=yeah
looooooooog=truncated
```

```
query =  '{' >> pair >>
          *(pair) > '}';
pair  =  '{' >> key >
          '=' >> val >>
          *('{' >> metakey >
          '=' >>
          metaval > '}') >
          '}';
```

```
Ni_node current = NULL;
while ((current =
          Ni_GetNextChild(root, current)) != NULL)
{
          Key *k = keyNew(0);
          keySetName (k,
                    Ni_GetName(current, NULL));
          keySetString (k,
                    Ni_GetValue (current, NULL));
          ksAppendKey (returned, k);
}
```

Limited Storage

- Hosts

- Simpleini

```
#/etc/hosts file

# a new comment


192.168.0.23        first_entry
192.168.0.24        second_entry
alias2 alias3 alias4 alias5
192.22.22.3         third_entry


# Markus follows


192.168.1.55        markus
```

```
user/simpleini/example1 = @NULL
user/simpleini/example_empty_string = @EMPTY
user/simpleini/example_null_value = @NULL
user/simpleini/example_text = @@text
user/simpleini/equal = %3Dequal%3D
```

- Null Values?
- Special Characters?

✔ Kommentare erhalten

✔ Reihenfolge erhalten

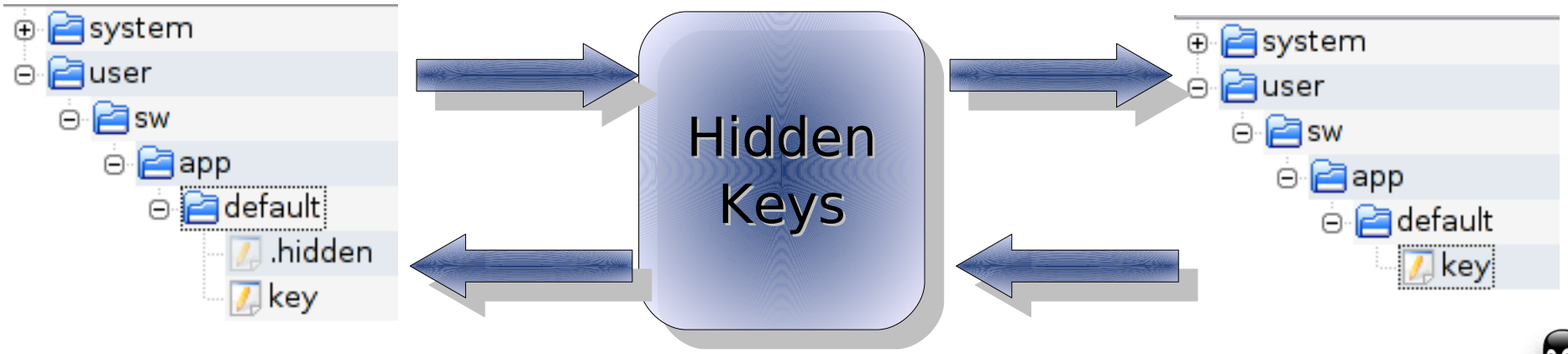
Filter Plugins

- ✔ Character Reduction

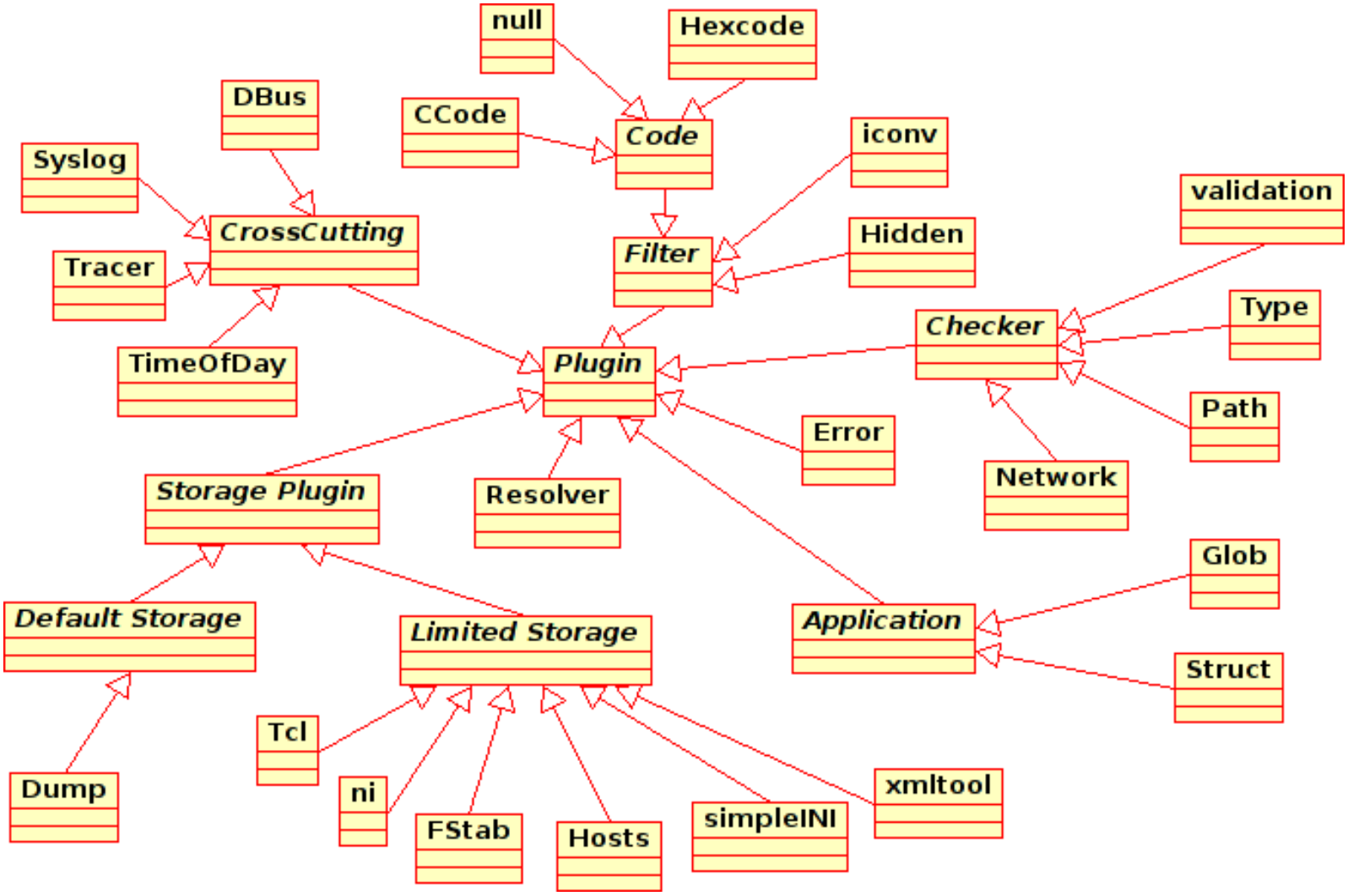
```
kdb set set user/ccode/key "'val          x'"
in storage: \'val\tx\'
```

```
kdb set user/hexcode/key value=abc xyz
in storage:  value%3Dabc%20xyz
```

- ✔ Iconv
- ✔ Application Specific Concerns



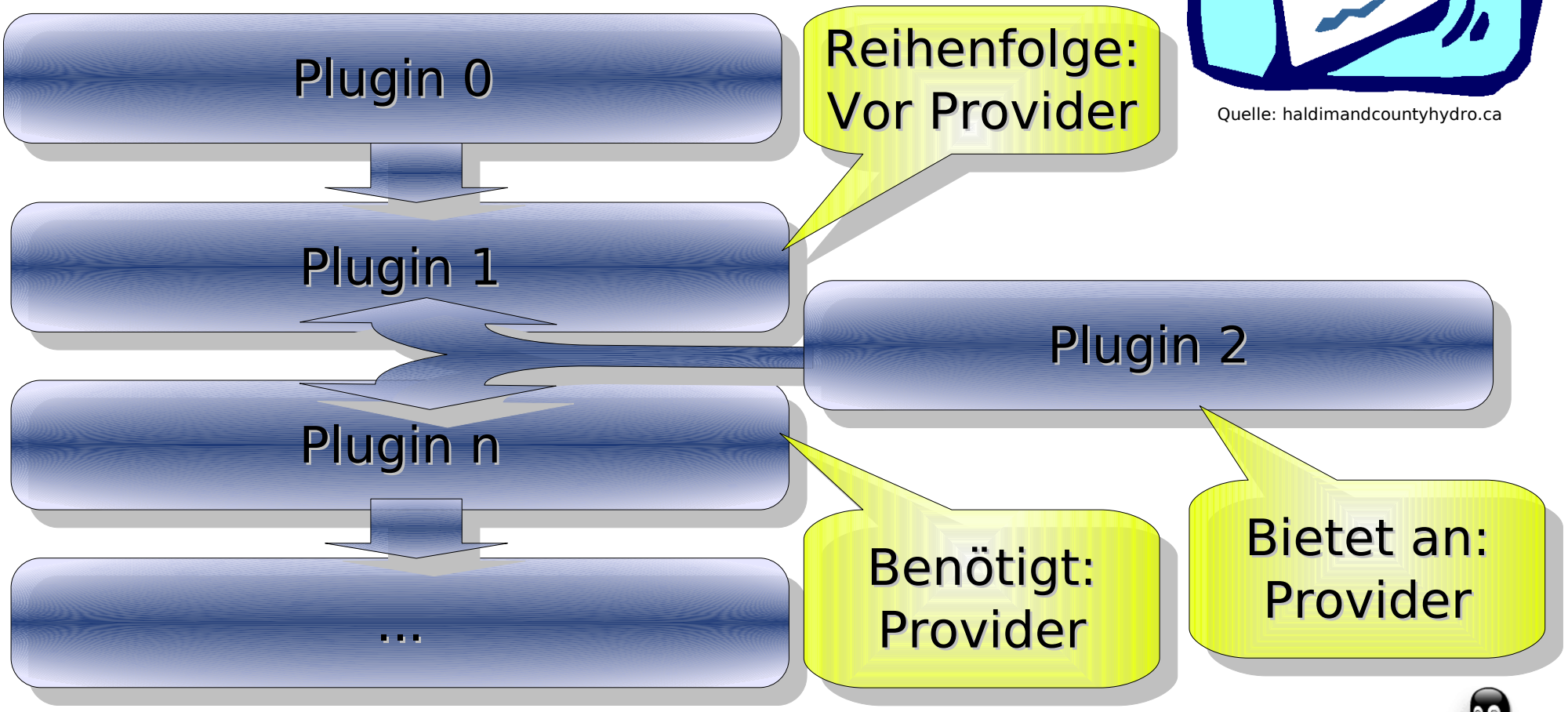
Plugins Übersicht



Kontrakte

- *Wer garantiert das?*
- *Richtigen Plugins?*
- *Richtige Reihenfolge?*

Quelle: haldimandcountyhydro.ca



Demo

Einsatz von Elektra

- ✔ *Embedded Systems*

BROADCOM®

Pengutronix.

- ✔ *Cross Plattform Software*

ουρανός

Oyranos

- ✔ *In Debian*

Ende

Vielen Dank für Deine Aufmerksamkeit!

Markus Raab <elektra@markus-raab.org>

Errors and Warnings

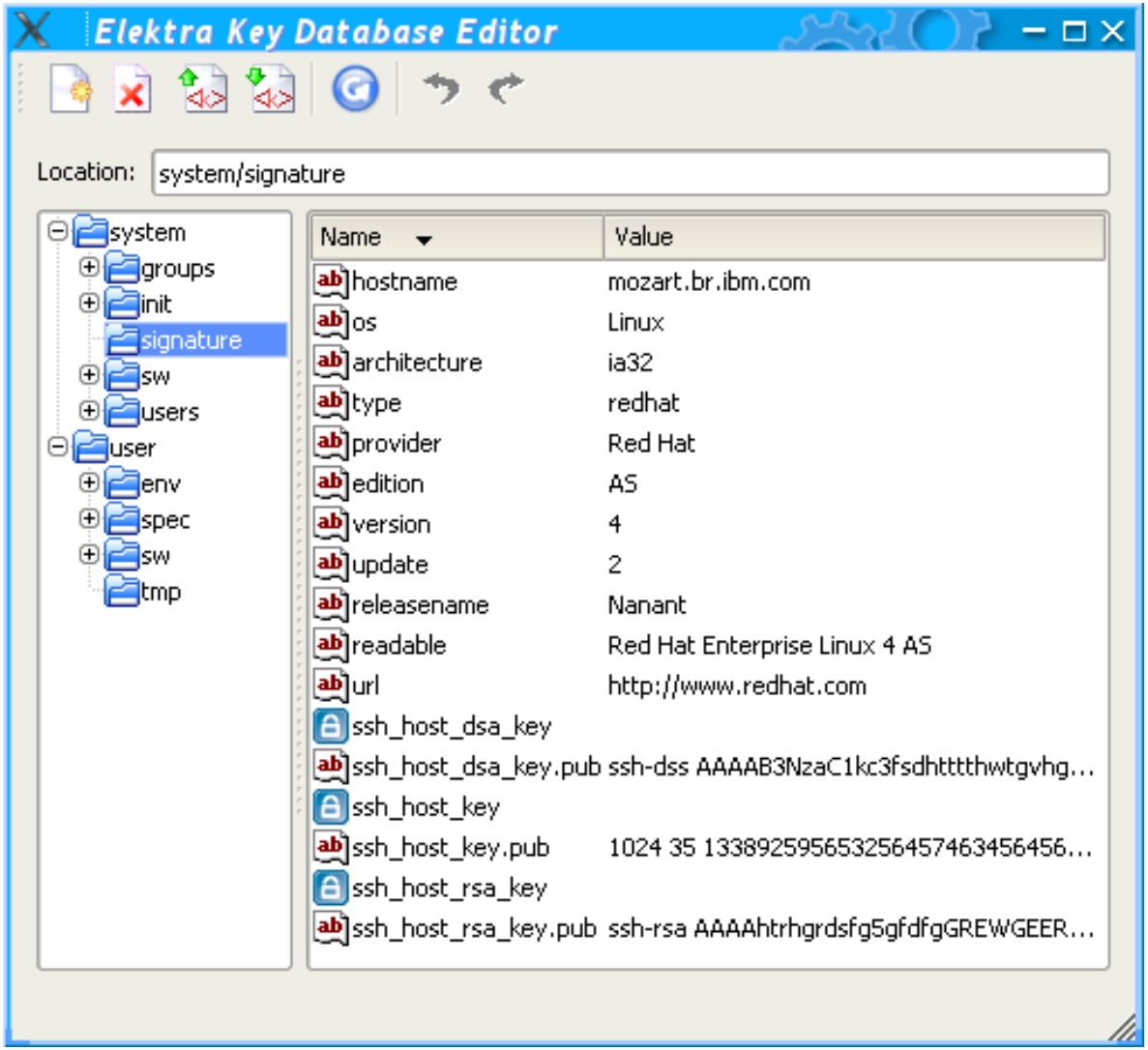
- ✔ *Volle Unterstützung von Fehlermeldungen*
- ✔ *Mittels Metadaten beliebige Texte*
- ✔ *Zentrale Datenbank*
- ✔ *(“Don't repeat yourself”)*
- ✔ *Unterstützung von Exceptions usw.*

Import, Export and Convert (TODO)

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

```
<?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 cleaver</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>
```

kdbedit: The Elektra GUI Edit (TODO)



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

Wenn alles versagt hast du hoffentlich das richtige Backend gewählt.

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

Einsatz von Elektra

- ✔ *Embedded Systems*

BROADCOM®

Pengutronix.

- ✔ *Cross Plattform Software*

ουρανός

Oyranos

- ✔ *In Debian*

Key Hierachie :: *system/...*

Administration mit Elektra

Linuxwochende 2010

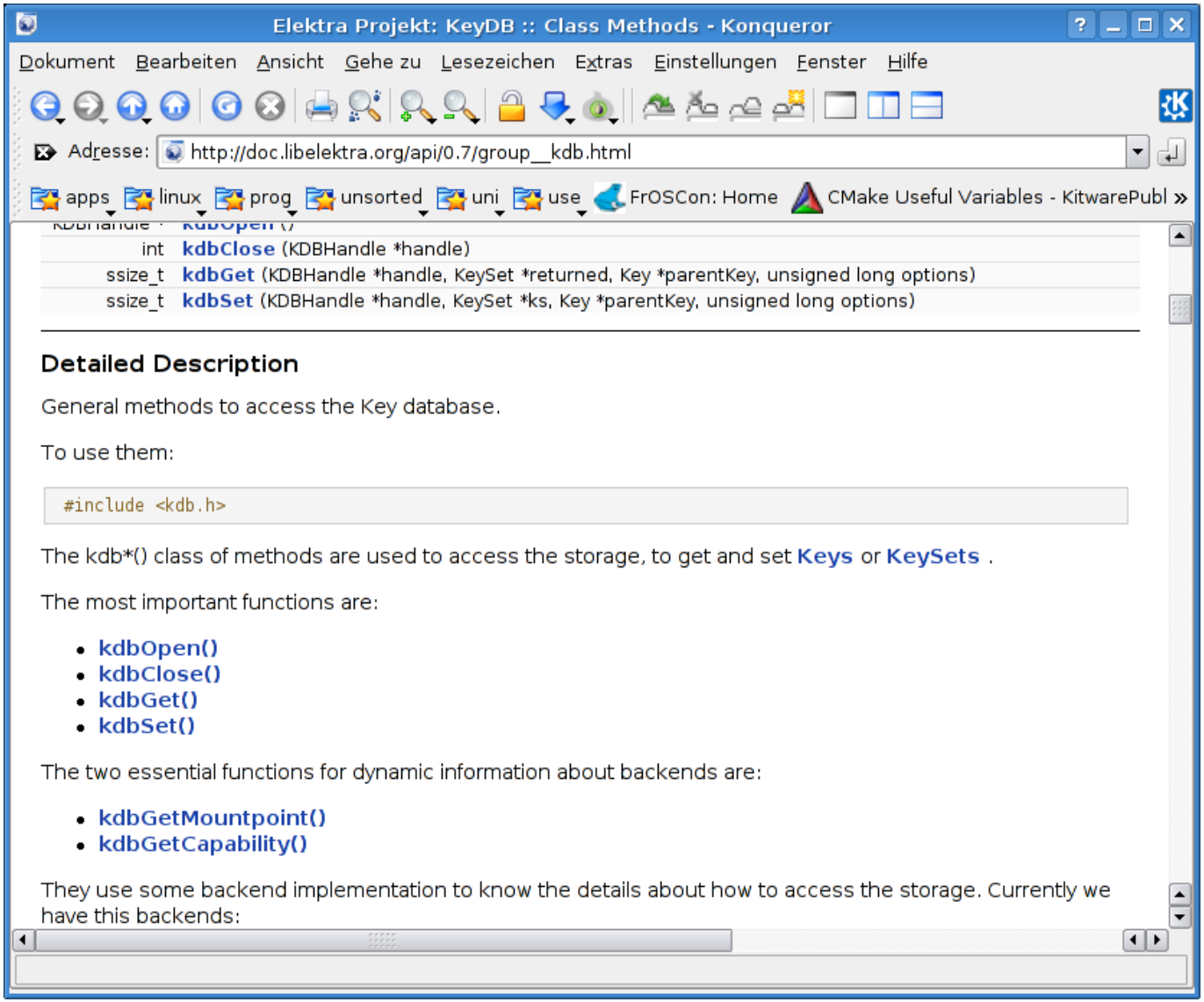
Markus Raab <elektra@markus-raab.org>

Qualität

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

The API is Fully Documented!

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

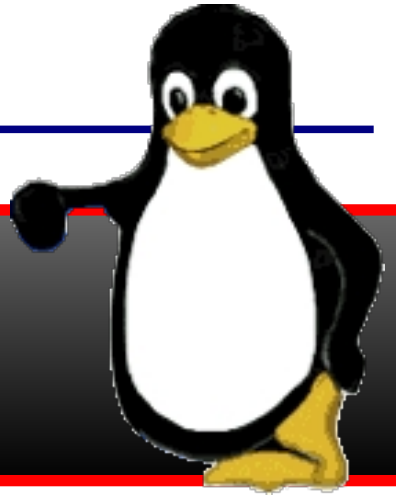


- ✔ API doc in Doxygen
- ✔ API man pages
- ✔ Viele Code Beispiele
- ✔ Programing Tutorial

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
```

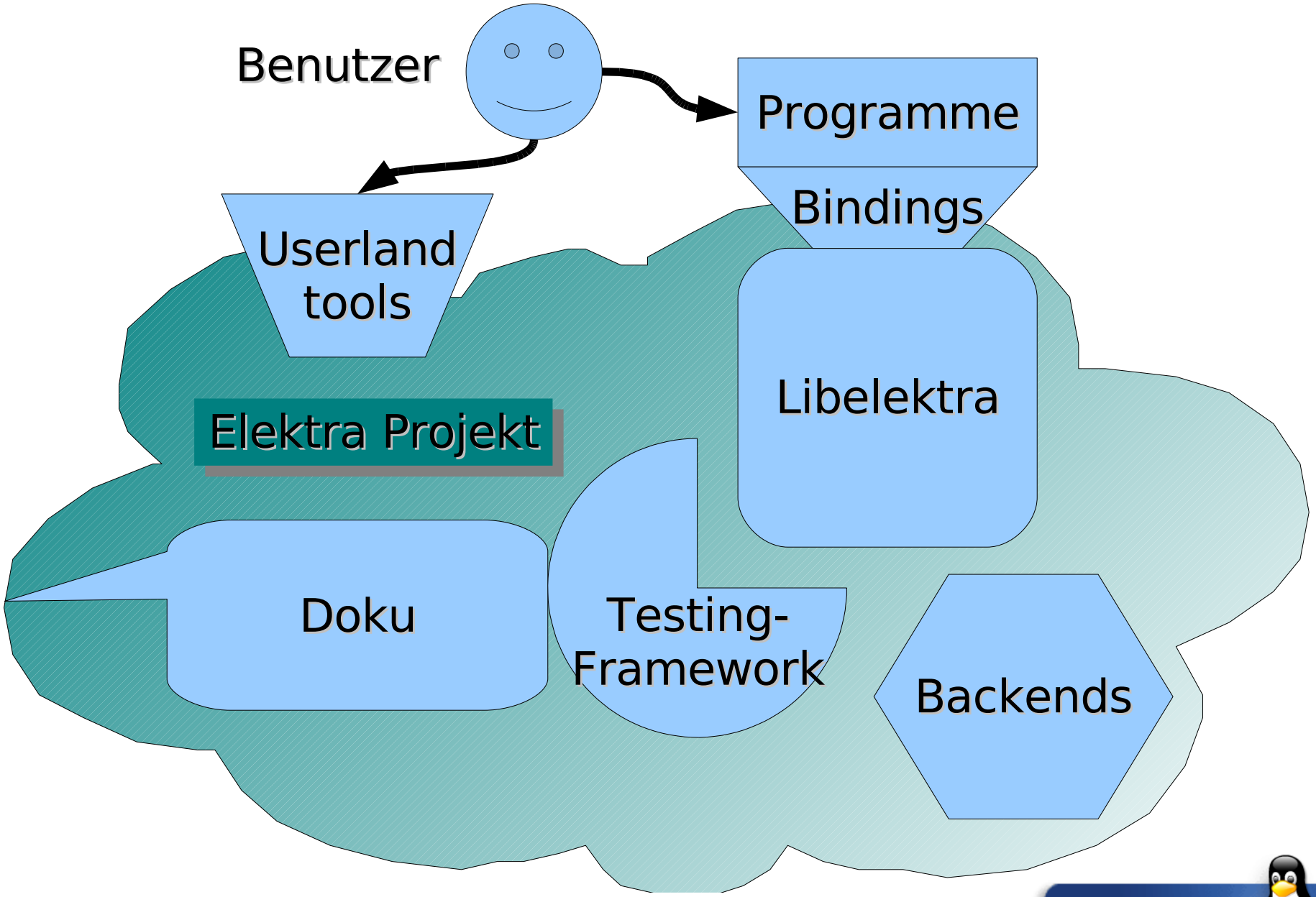
Elektra ist leicht . . .



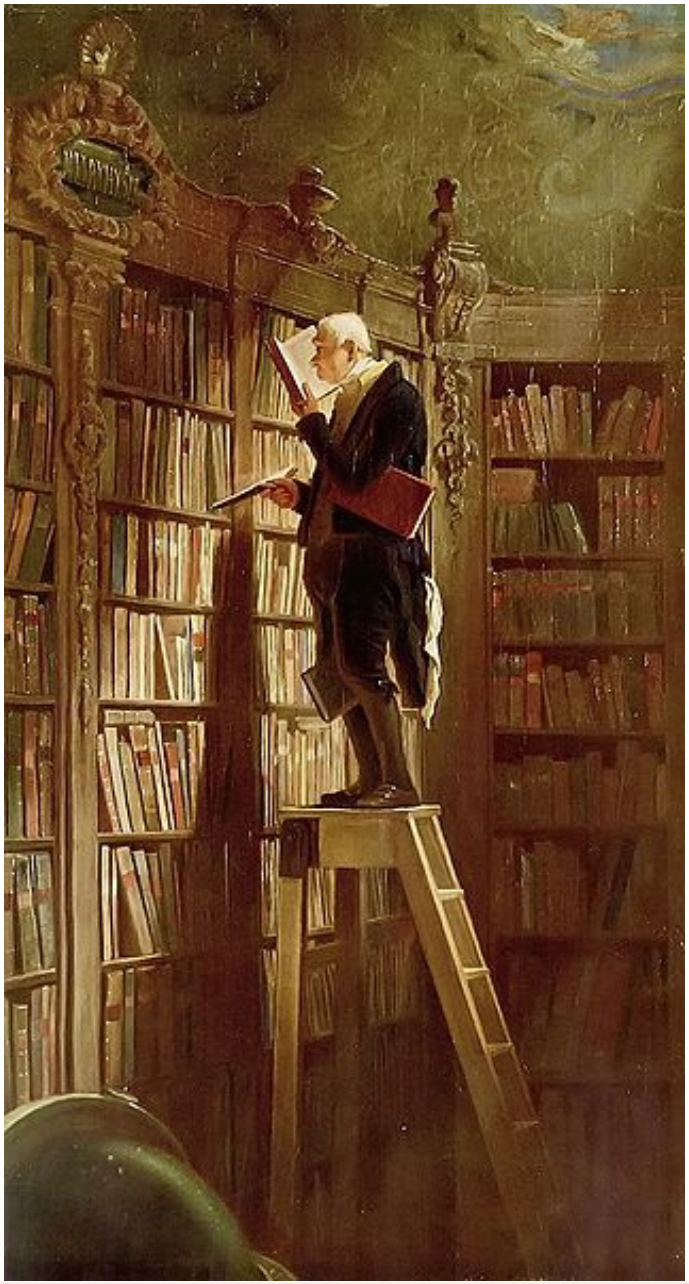
```
bash$ ldd /lib/libelektra.so
    libc.so.6 => /lib/tls/libc.so.6
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2
```

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

Ecosystem


Benutzer
Programme
Bindings
Userland tools
Elektra Projekt
Libelektra
Doku
Testing-Framework
Backends

Elektra als Bibliothek



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

Ee Ll Ee Kk Tt Kr Aa

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 dumpfreq;
        unsigned short passno;
};
```

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

Related Work

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

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
```

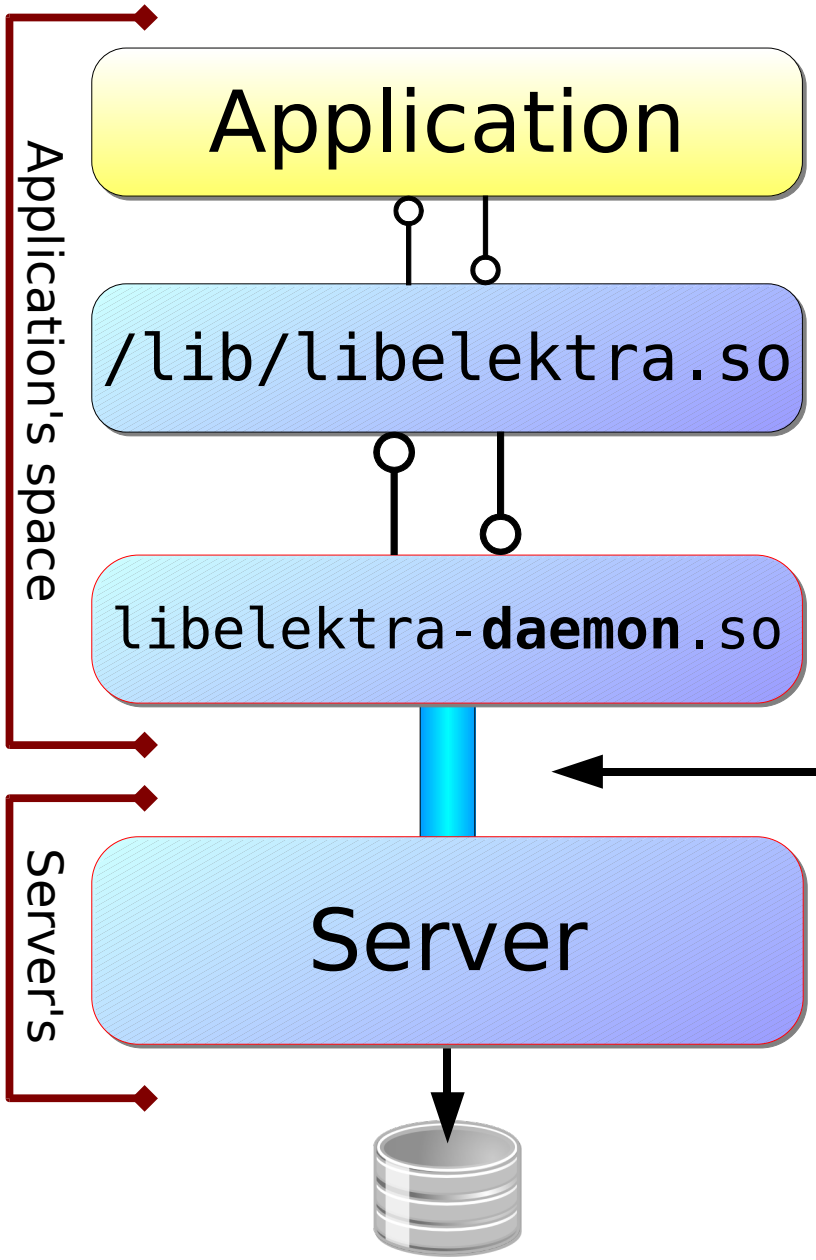
```
service saft
{
        socket_type = stream
        protocol = tcp
        wait = no
        user = root
        server = /usr/sbin/sendfiled
        port = 487
        disable = no
}
```

- ✔ Unterschiedliche Bedürfnisse
- ✔ Identität von Software

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

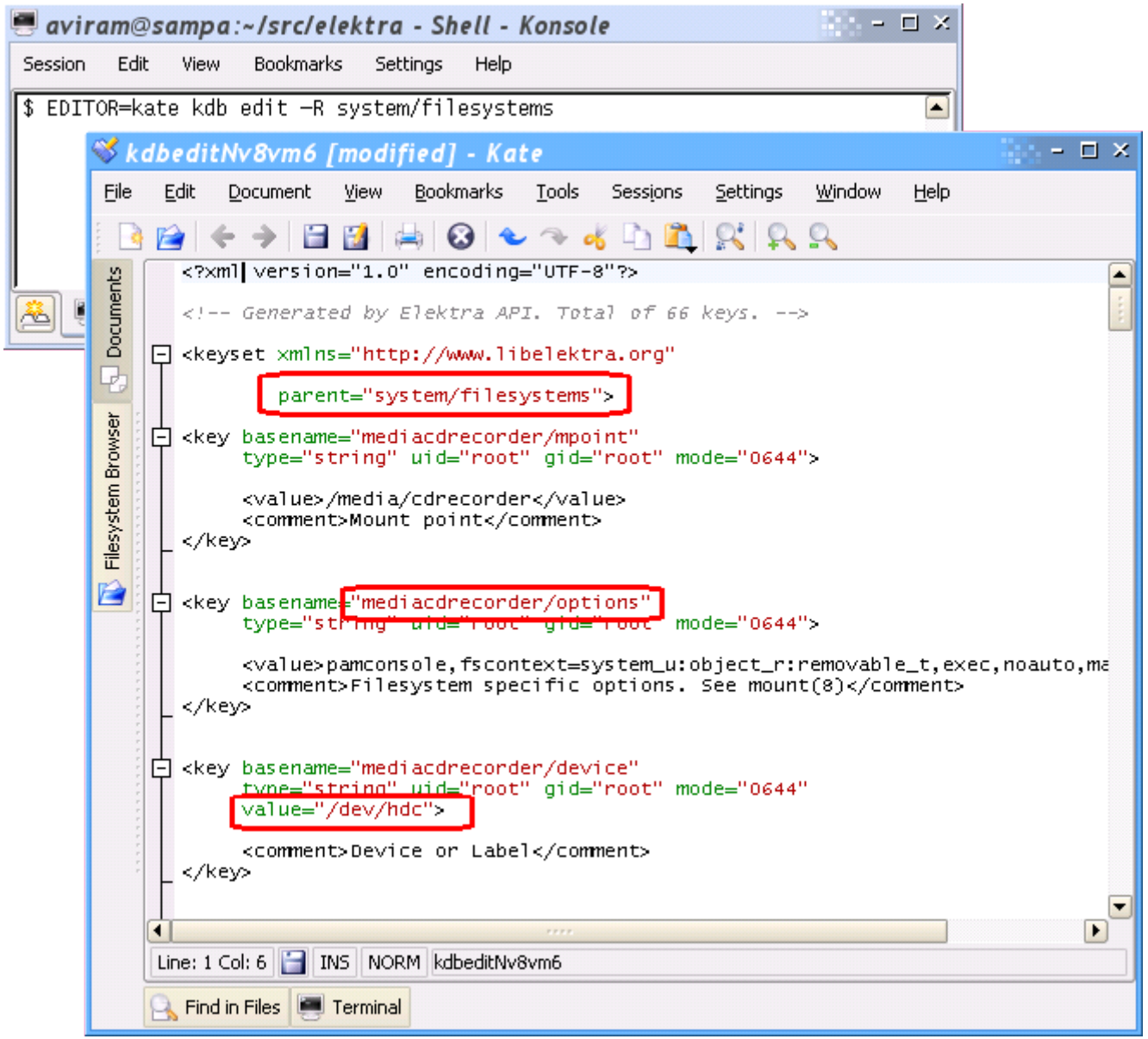


➔ ***Client-Server Architecture***

- Kommunikation zum Server über *libelektra-**daemon**.so*
- Der Server selber verwendet auch *libelektra.so* um key/value zu speichern.
- single point of failure

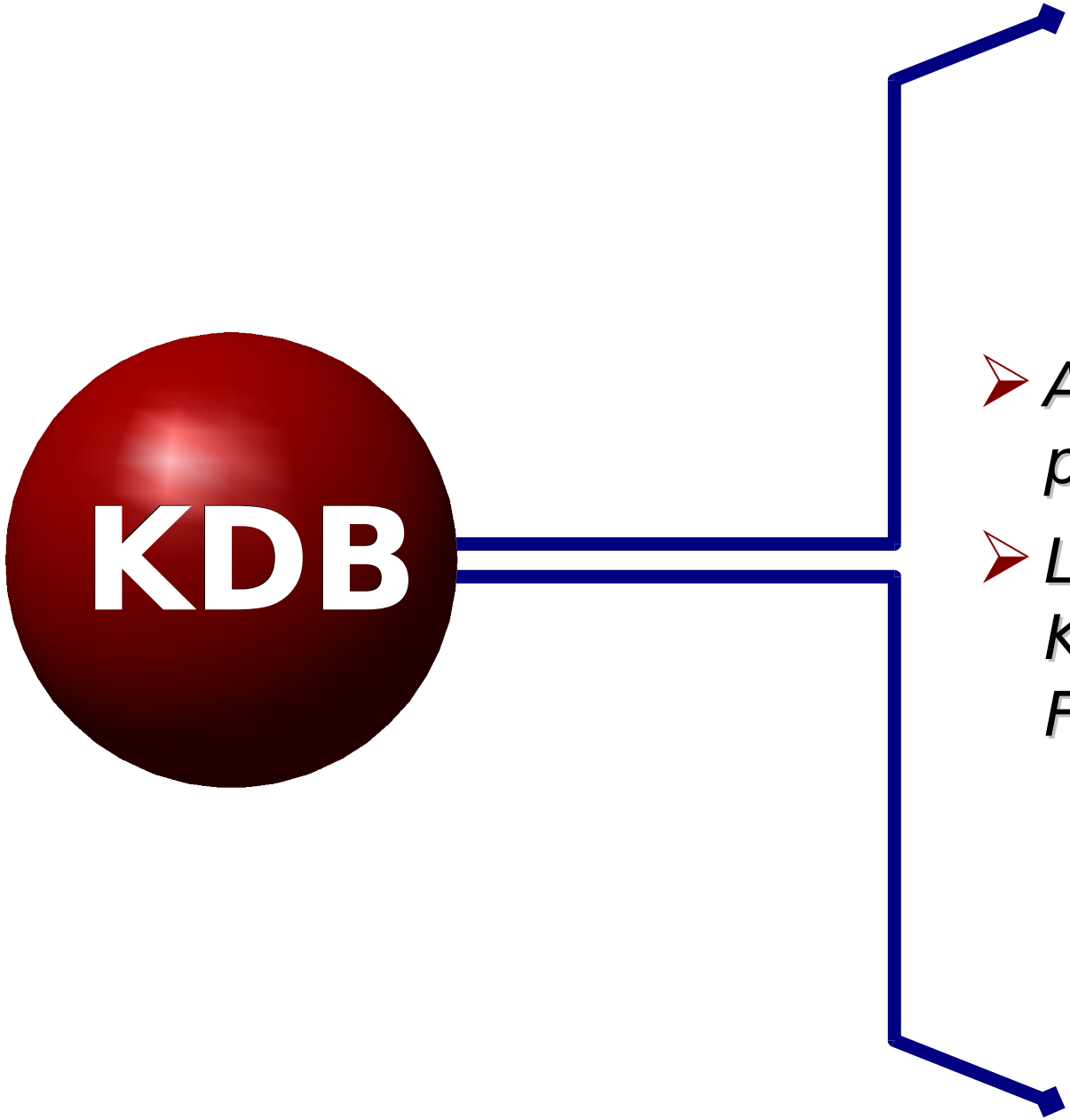
Protokol über unix domain sockets

Editieren von key/value pairs (0.7)



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

Key Data Base



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

Keyset Eigenschaften

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

Fertig um jetzt benutzt zu werden.

- ***Vorhanden***
  - ✔ C
  - ✔ C++
  - ✔ Shell
  - ✔ XML
- ***War mal da . . .***
  - ✔ Python
  - ✔ Scheme
  - ✔ Ruby
  - ✔ Java

The Elektra Initiative needs YOU !

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

Related Work

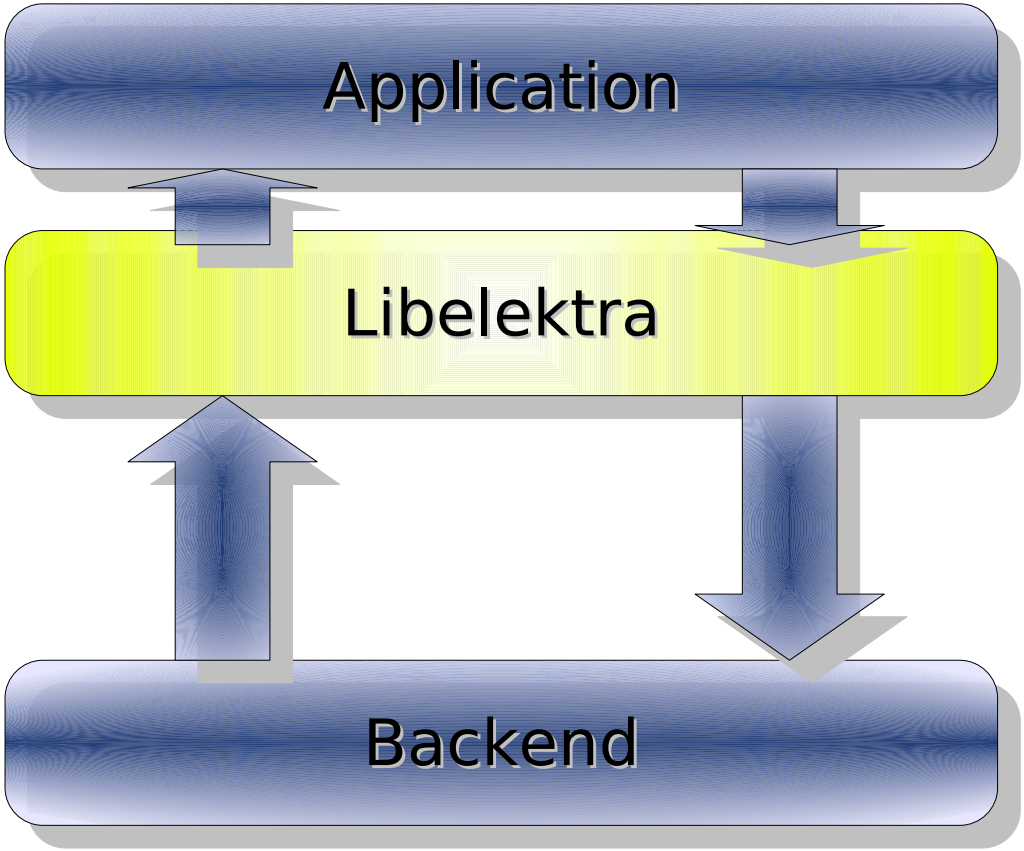
- *Elektra with Mounting*
- *Uniconf (with Daemon)*
- *Debconf (supports stacking)*
- *Augeas (Linux Configuration Files only)*
- *Pam (Error Conditions)*

Modulare Backends für Elektra

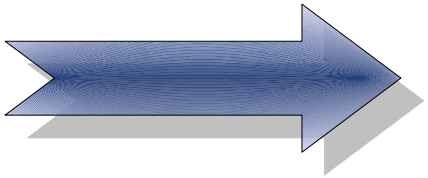
Linuxwochende 2010

Markus Raab <elektra@markus-raab.org>

Vorige Situation



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



Modularität in Backends?

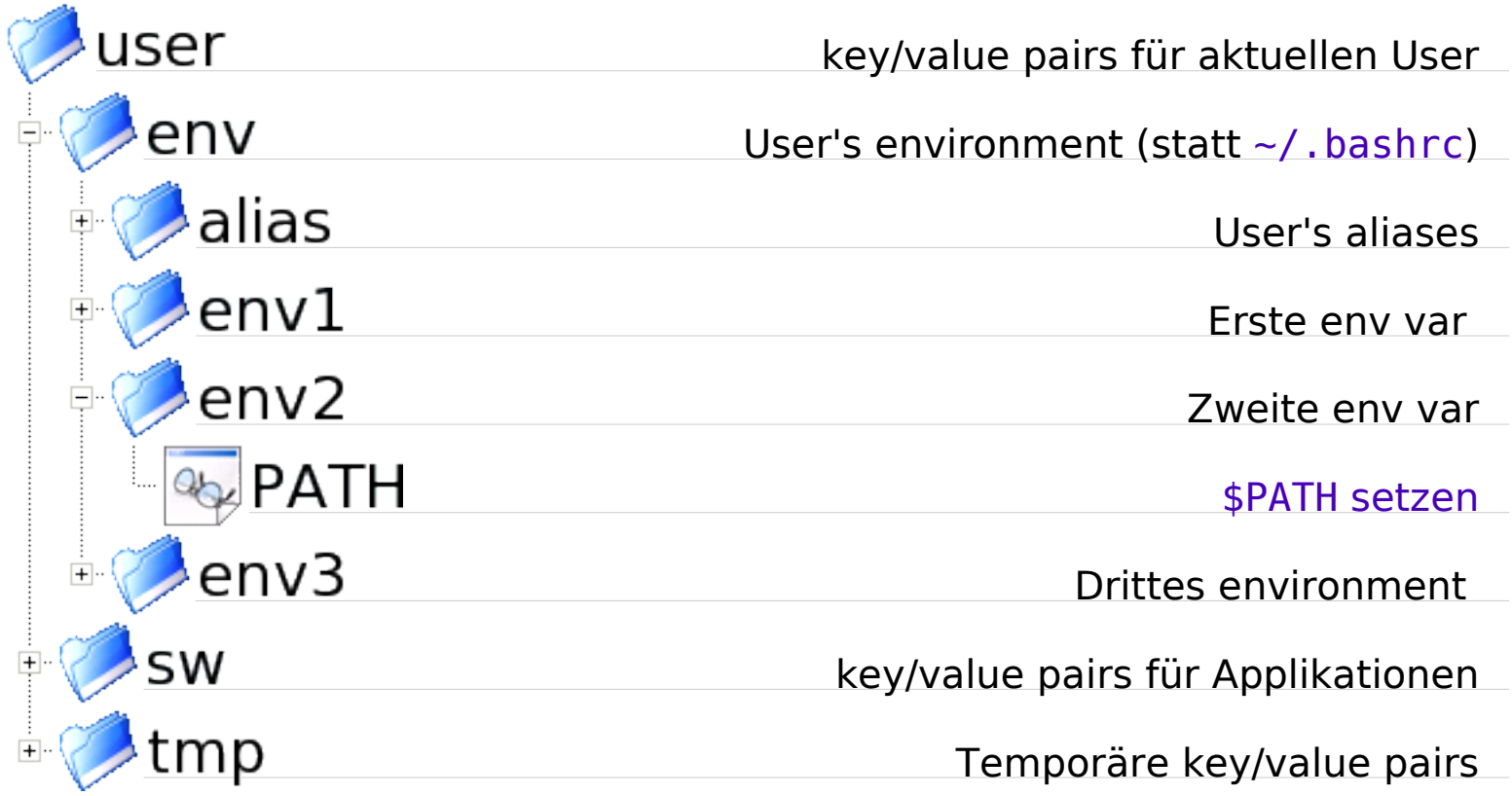
Key Namen

Beispiel für Key Name	Value
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/De ice/Videocard0/Driver	radeon
system/sw/apa	/var/www/site1.com
user/env/alia	ls -Fh
user:valeria/sw/regedit/gui/width	747

Dieser kleiner Punkt macht die ganze Hierachie inaktiv?

Key Hierachie :: *system/...*

Key Hierachie :: *user/...*



Kaskading

Beispiel für Key Name	Value
system/filesystems/boot/device	/dev/hda1
/shortcut/menu/open	Strg+O
sy…tem/net/ISP/AOL/phone	555-2132
	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

Durch diesen Slash wird der richtige Schlüssel automatisch gesucht