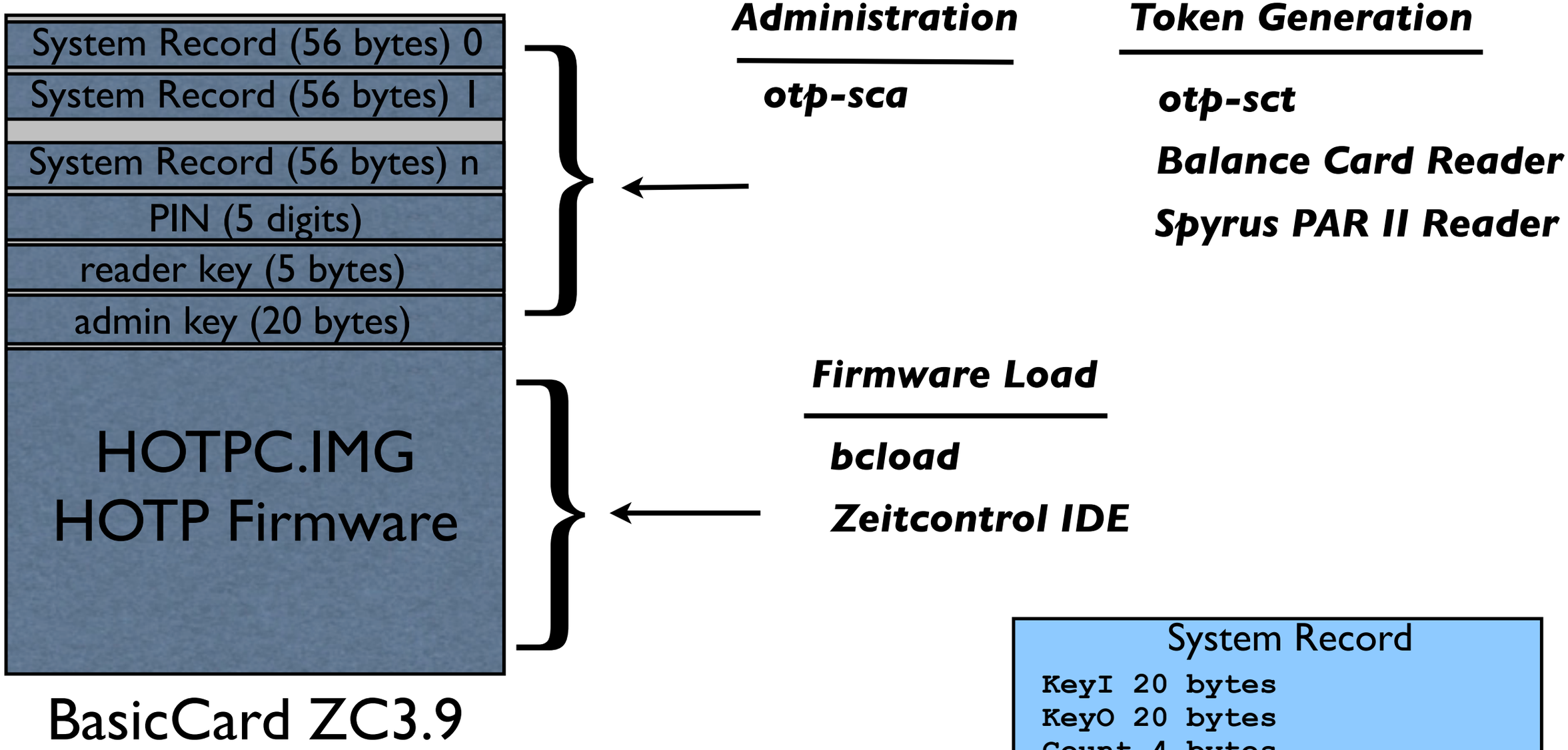


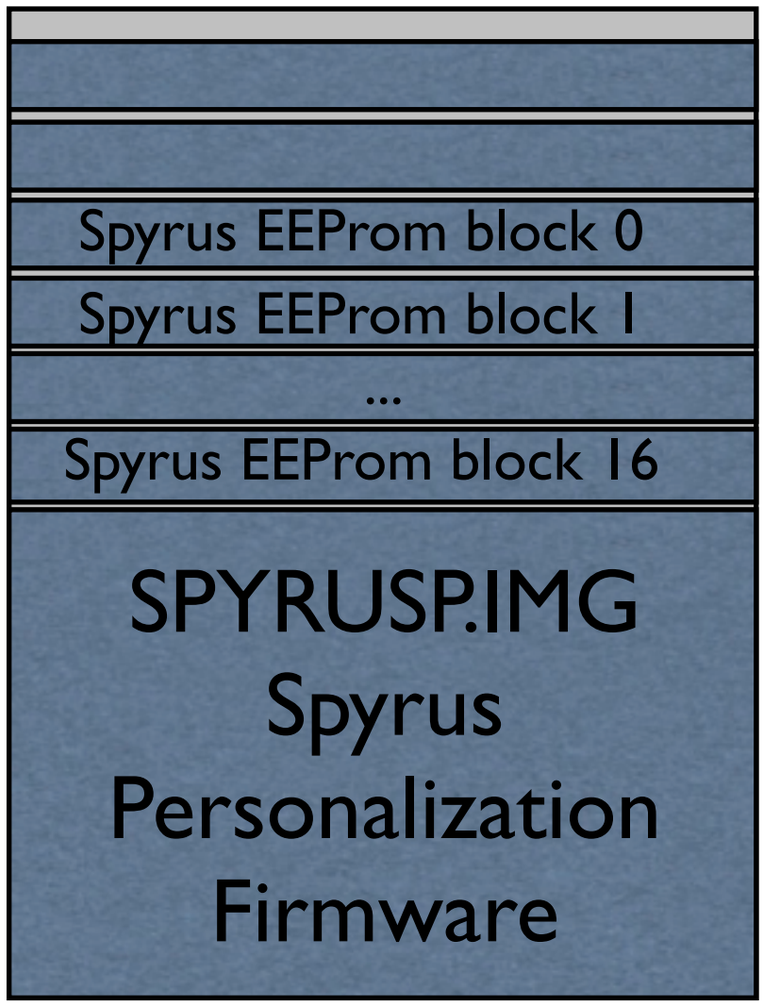
System Record

```
KeyI 20 bytes
KeyO 20 bytes
Count 4 bytes
Hostname/flags 12 bytes
-- 12 character hostname
   12 flag bits (high bit)
```

```
SC computes HOTP, stores keys and system names

otp-sca programs SC with keys
display HOTP with Spyrus Reader, Balance Reader, or SC reader and otp-sct.

bcload to program SC firmware with SPYRUSP
```



BasicCard ZC3.9

Administration

otp-sca

Firmware Load

bcload

Zeitcontrol IDE

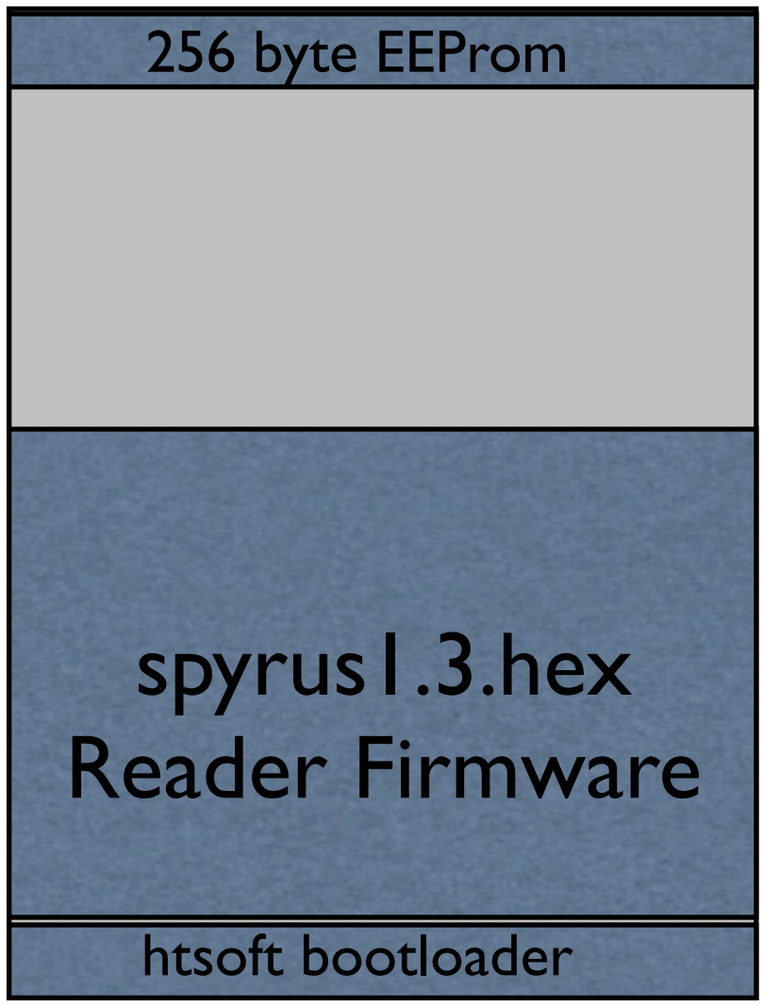
SC is used as transport for EEPROM data

otp-sca programs SC with EEPROM blocks
Spyrus reader copies blocks to EEPROM from SC

bcload to program SC firmware with SPYRUSP

EEProm Defaults

EE_MAGIC	:maf:
EE_READER_KEY	:00000:
EE_CALC_MSG	:OARnet:2009 :
EE_L1GREET	: OARnet :
EE_L2GREET	:PIN: :
EE_L1MAIN	: OARnet :
EE_L2MAIN	: Verified :
EE_CHALLENGE	:Challenge: :
EE_L1LOCKED	:10 Failures :
EE_L2LOCKED	:Card Locked :
EE_L1ACCESS_DENY	: Access :
EE_L2ACCESS_DENY	: Denied :
EE_NOHOSTS	: No Hosts :
EE_L1NEWPIN	:Set New PIN :
EE_L2NEWPIN	:NewPIN: :
EE_L3NEWPIN	:Again: :
EE_PINCHANGED	:PIN Changed :
EE_NOCARD	:No Card :
EE_TRYHARDER	:Try Harder :



Spyrus PAR II
PIC16F877

EEProm for display strings, such as "OARnet" and reader-key*

Programmed with Spyrus Personality Smart Card.
Functional defaults in program flash.

htsoft-downloader firmware loader
or
pic-downloader firmware downloader

Factory supplied bootloader for loading firmware via serial port with downloader application.

***reader-key is sent by the Spyrus reader to the smart card. If the smart card system is configured to require the reader key, the smart card reader-key must match this key before generating a HOTP. This is weak authentication to allow us to require using the Spyrus reader, and not a PC connected smart card reader to generate tokens.**

Spyrus HOTP Generation

PAR II SDK provides communication between PIC and co-processor

User enters PIN, selects system name. PAR II displays HOTP generated from Smart Card

PIC16F877

Co-processor

BasicCard

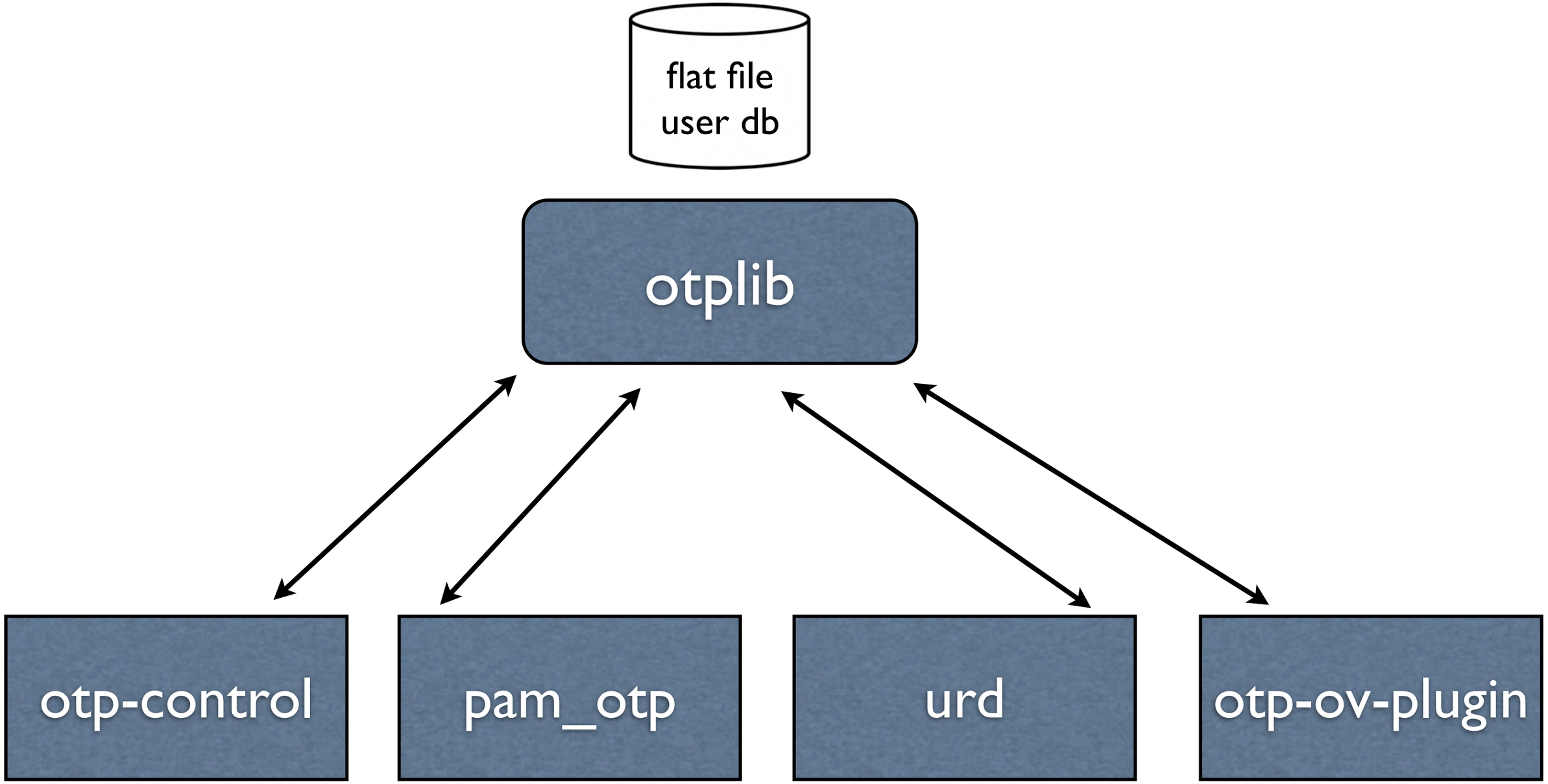
Open source HOTP terminal software. Binary included, requires PAR II SDK & HI-TECH C compiler to build source.

Offloads pinpad, LCD, and SC communications from PIC16F877

HOTPC.IMG
Per System {Hostname,Key,Count} HOTP Calculation

SPYRUSP.IMG
EEPROM customization card.

Run-time programmable EEPROM to customize menu items without rebuilding from source.



OTP DB management. Add users, create keys, etc.

PAM module for ssh, login, etc

Micro RADIUS server for network authentication

OpenVPN authentication plug-in

Spyrus

Hardware
50+ keys w ZC3.9
5 digit PIN
40 bit HEX HOTP
6-10 digit base10 HOTP
Reprogrammable

Production deployment. Advantage is multiple reprogrammable keys per readers + PIN.

Balance Card

Hardware
1 key
No PIN
40 bit HEX HOTP
6-10 digit base10 HOTP
Reprogrammable

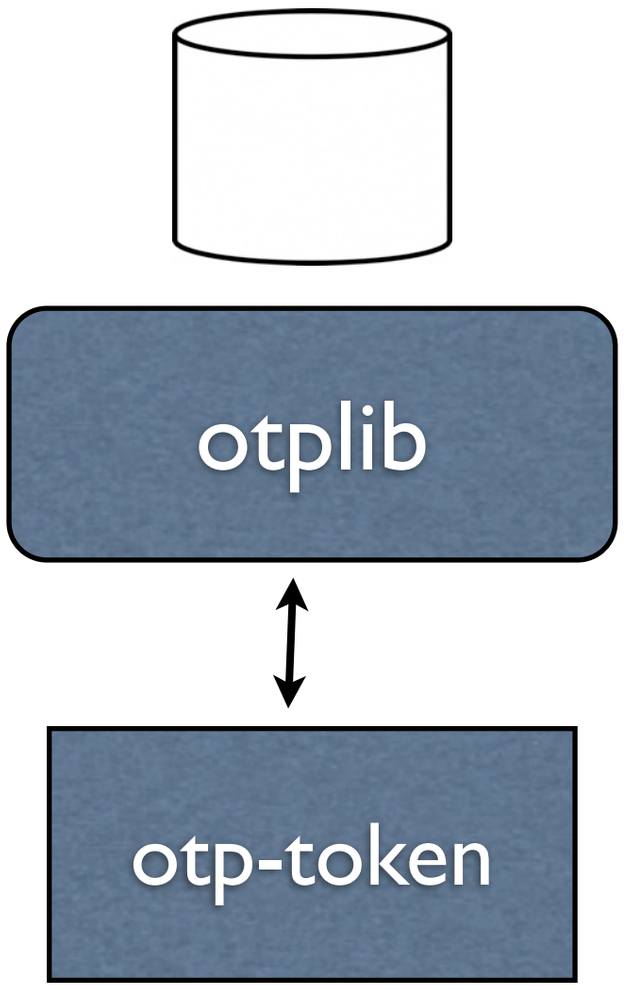
Did not work well in small deployment, balance readers easily broken. Purpose-built HOTP generators available now which would work better at similar cost.

PC Card Reader

Software PC client + Smart Card
50+ keys w ZC3.9
5 digit PIN
40 bit HEX HOTP
6-10 digit base10 HOTP
Reprogrammable

Requires otp-sct Linux/BSD/MAC
Not really two-factor anymore as anyone with access to system can generate HOTP once SC is in.

Soft Token



Software
Unlimited keys
Passphrase protected keys
40 bit HEX HOTP
6-10 digit base10 HOTP

Not two factor

Others...

Java client, IPhone APP, hardware tokens
Should work with ootp

Any PCSC-Lite supported reader

ACR30S chipset based Smart Card Reader

USB/RS232 Interface

RS232 Interface

pcscd (PCSC-Lite)

embedded acr30s or PC/SC Driver

otp-sca

otp-sct

bcload

Smart Card Admin. Load Systems, change PIN, etc.

Smart Card Terminal. End-user program to generate HOTP with PC connected SC reader

BasicCard Firmware Loader (Unix).

Smart Card database files

.card files

r/w **otp-sca**

w **otp-control**

record format: index:count:hostname:key
ASCII HEX fields
1 record per line

Index: 0..254. SC index to system. Spyrus menu will display in ascending order.
Count: HOTP Count
Hostname: Menu item for display on Spyrus Reader
Key: 160 bit HOTP Key

High bit of hostname characters used as flag bits.

```
0:   CHALLENGE  if set then prompt for count before generating HOTP
1:   READERKEY  if set require valid reader key before generating HOTP
2:   FMT        if set display HOTP in decimal
3-7: RESERVED
8:   FORMAT0    0,1=HEX40   2=DEC31.6  3=DEC31.7
9:   FORMAT1    4=DEC31.8   5=DEC31.9  6=DEC31.10
10:  FORMAT2    7=DHEX40    8-15 RESERVED
11:  FORMAT3
```

Back-End User database files

$OTPDB/d/<username> files

r/w **otp-control, pam_otp, otp-openvpn, urd, any otplib auth client**

record format: version:user:key:status:format:type:flags:count_cur:count_ceil:last
ASCII HEX fields
1 record per file (dump/import in otp-control will allow multiple records per file)

Version:	**1**
user:	**username**
key:	**160 bit HOTP key**
status:	**Account status {active, inactive, disabled}**
format:	**Account format {hex40, dec31.6, dec31.7, dec31.8, dec31.9, dec31.10, dhex40}**
type:	**Account OTP Type {HOTP}**
flags:	**Account flags {display-count}**
count_cur	**Current count**
count_ceil	**Count ceiling**
last	**Last login time in Unix seconds**