

# Gluing Together Desktop Crypto

Stef Walter



Three steps to

# Gluing Together Desktop Crypto

Stef Walter





**PKCS11**

# Three steps....



Store keys and certificates interoperably



Make consistent trust decisions



Refer to keys and certificates in a standard way





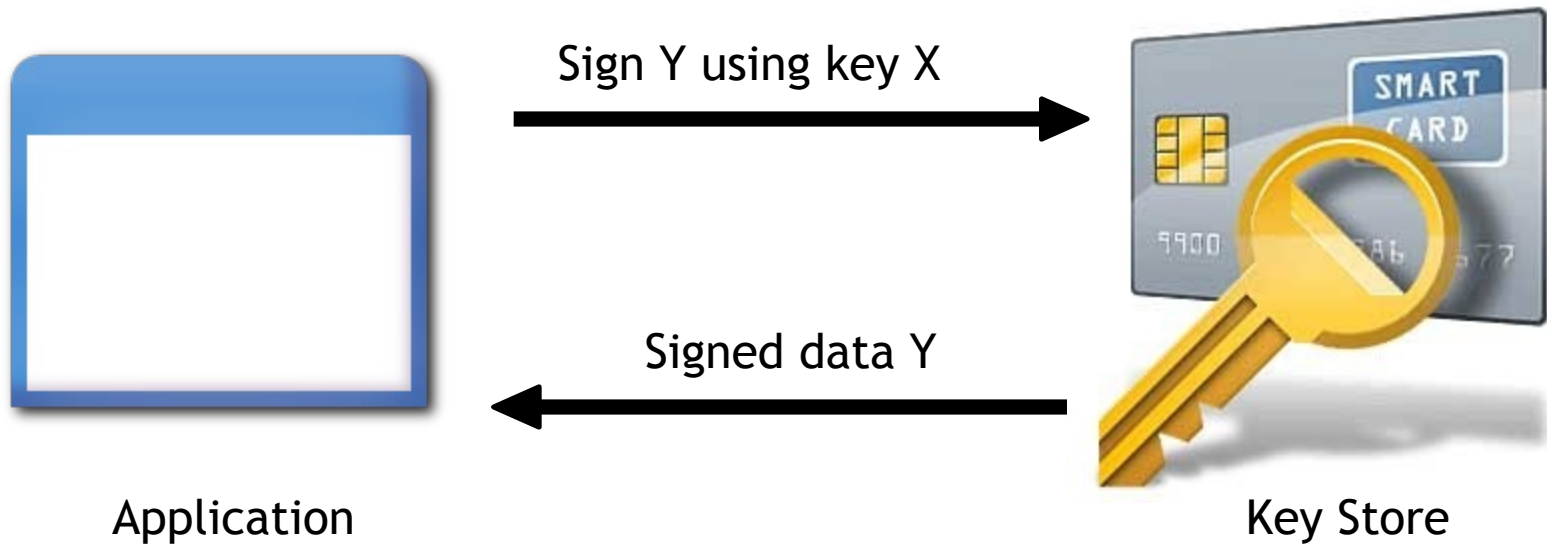
Spock warns: **“Be careful with glue”**

# Key Storage



# What is a key store?

What makes it different?

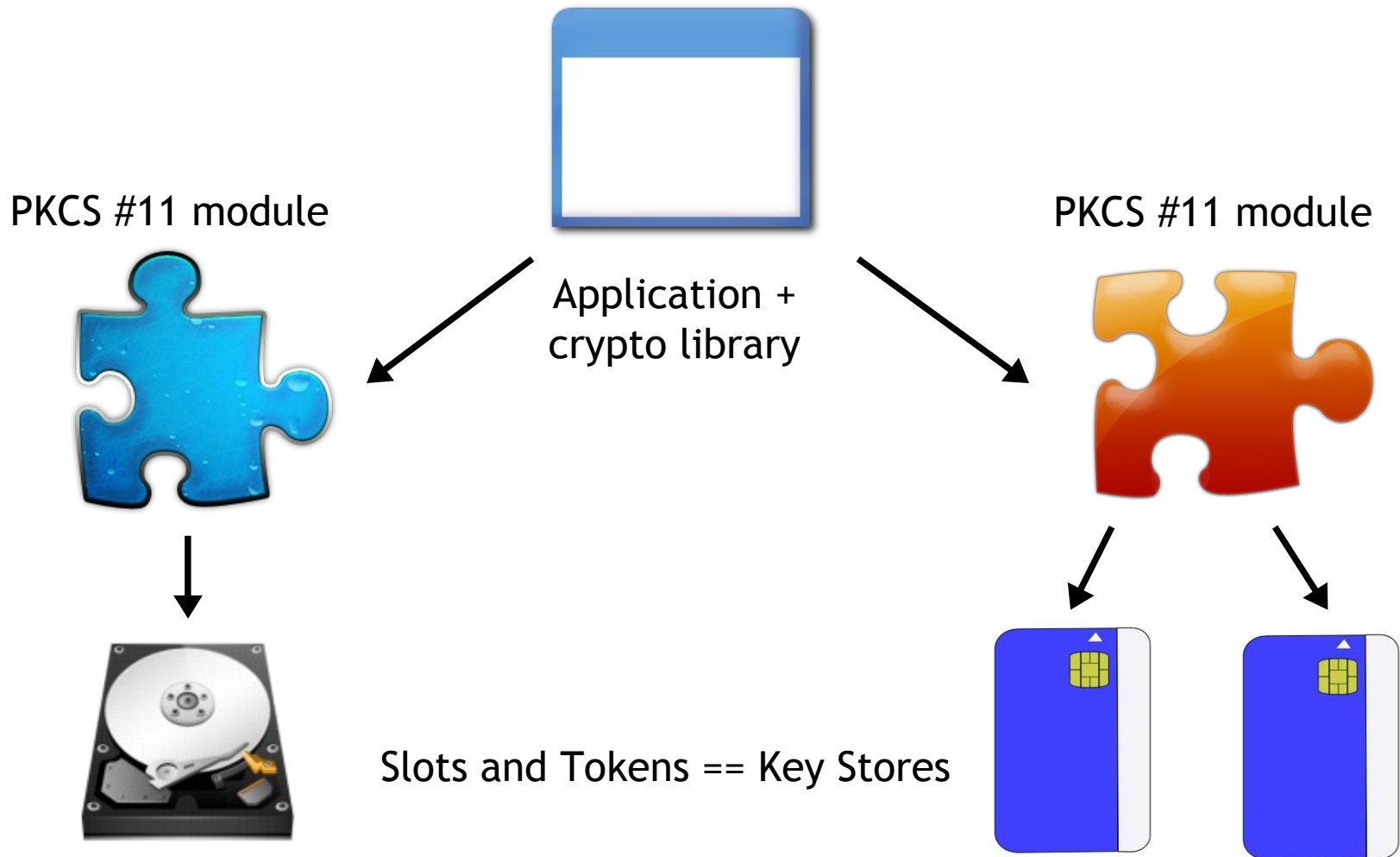


# PKCS#11





# PKCS #11 Concepts



# Support for PKCS#11

## Decent Support:

- GnuTLS
- GNOME Keyring
- Java (SUN)
- Mozilla's NSS
- OpenSC
- OpenSSH
- OpenVPN
- QCA (QT)
- TrueCrypt

## Work in Progress

- GLib
- OpenSSL

## Patches Available:

- GnuPG

... and many others



# p11-kit: Solves PKCS#11 on the Desktop

1. Using the same PKCS#11 modules more than once in the same process.
2. Configuration: A standard way to lookup which modules are installed and enabled.

... and other handy things ...



# PKCS#11 Module Configuration

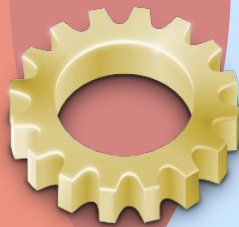
**System**  
/etc/pkcs11



Global Config  
pkcs11.conf

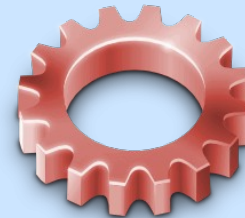


Smart Card Driver  
modules/smart-card



Keyring module  
modules/gnome-keyring

**User**  
~/.pkcs11



Power user module  
modules/kssl-storage



User Config  
pkcs11.conf

# PKCS#11 Configuration Lockdown

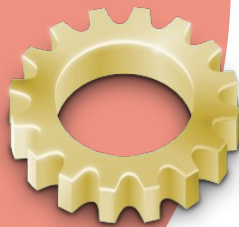
**System**  
`/etc/pkcs11`



Global Config  
`pkcs11.conf`



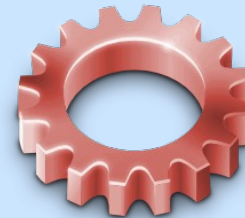
Smart Card Driver  
`modules/smart-card`



Keyring module  
`modules/gnome-keyring`

# PKCS#11 Module Configuration

**User**  
~/.pkcs11



Power user module  
modules/kssl-storage



User Config  
pkcs11.conf

# Library: p11-kit

<http://p11-glue.freedesktop.org/p11-kit.html>



# Three steps....



Store keys and certificates interoperably



Make consistent trust decisions



Refer to keys and certificates in a standard way





# 'Trust'



# 'Trust'

What does that even



# Trust Assertions

Each Trust Assertion makes a positive or negative assertion about level of trust in a subject.



# Trust Assertions



**Subject**

**Level of Trust**

**Purpose**

# eg: Certificate Trust Anchor

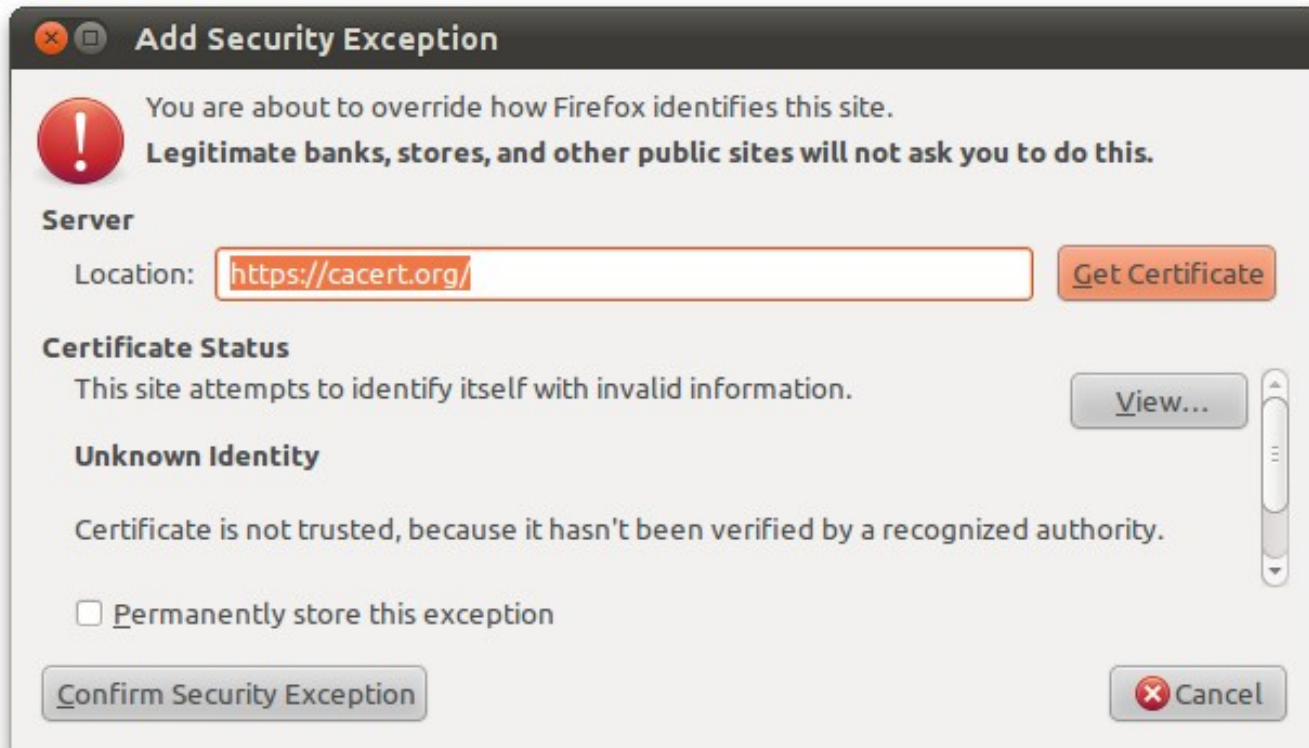


**Certificate**  
Subject

**Introducer**  
Level of Trust

**Server Auth.**  
Purpose

# eg: Pinned Certificate Exception

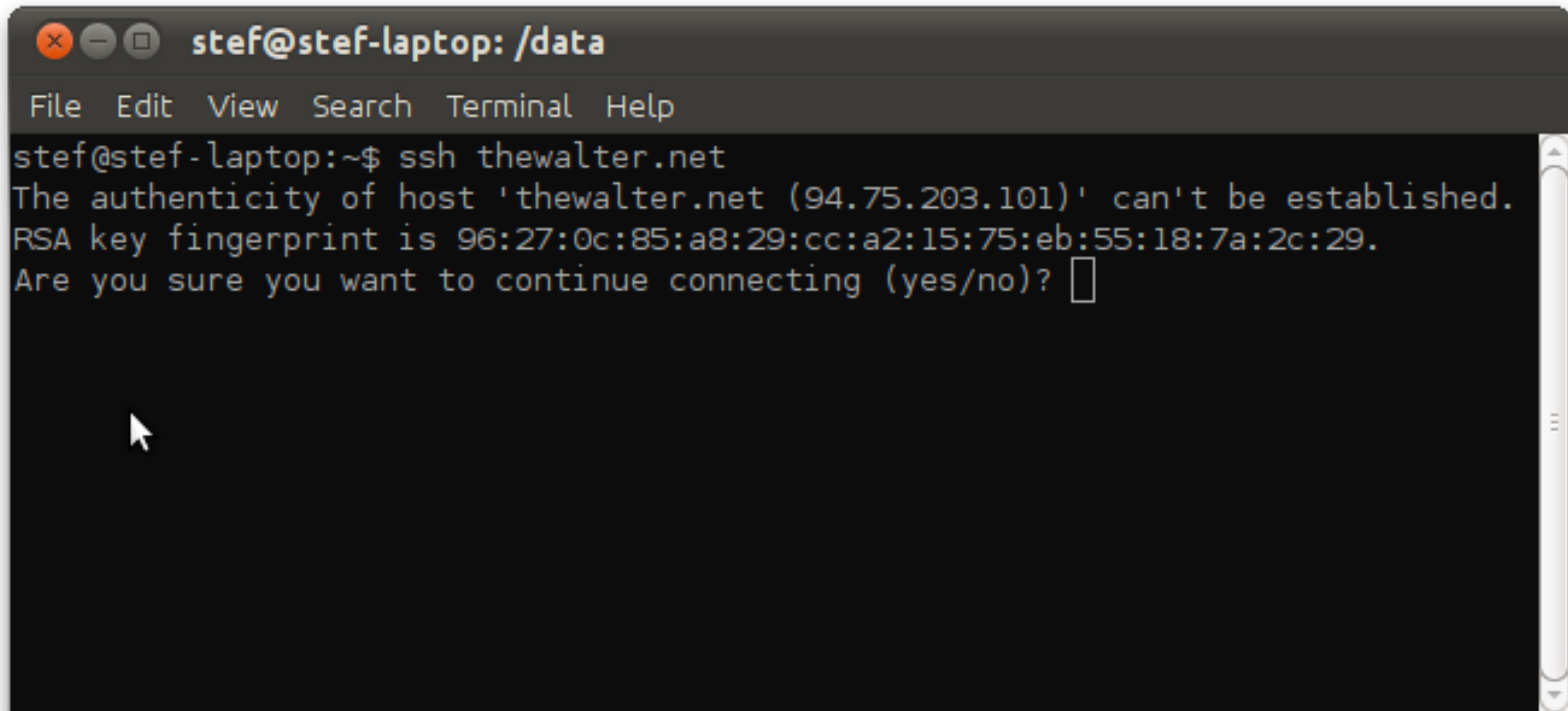


**Certificate**  
Subject

**Trusted**  
Level of Trust

**Server: ca.cert.org**  
Purpose

# eg: SSH Known Host



```
stef@stef-laptop: /data
File Edit View Search Terminal Help
stef@stef-laptop:~$ ssh thewalter.net
The authenticity of host 'thewalter.net (94.75.203.101)' can't be established.
RSA key fingerprint is 96:27:0c:85:a8:29:cc:a2:15:75:eb:55:18:7a:2c:29.
Are you sure you want to continue connecting (yes/no)?
```

**Public key**  
Subject

**Trusted**  
Level of Trust

**Host: thewalter.net**  
Purpose

# eg: Certificate Revocation List



**Serial+Issuer**  
Subject

**Distrusted**  
Level of Trust

**Any**  
Purpose



# Spec: Trust Assertions in PKCS#11

<http://p11-glue.freedesktop.org/trust-assertions.html>



# Three steps....



Store keys and certificates interoperably



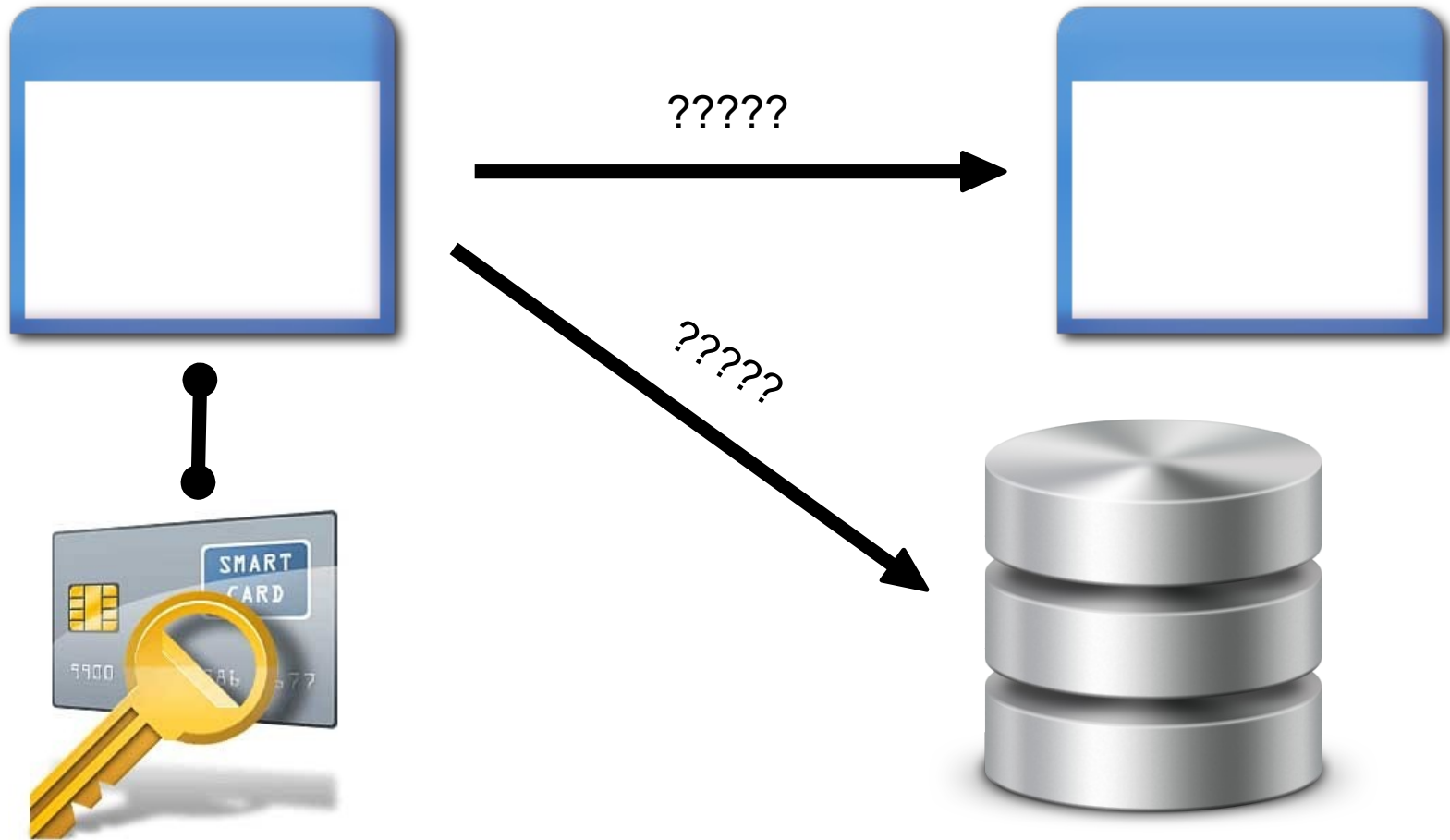
Make consistent trust decisions



Refer to keys and certificates in a standard way



# How do you refer to Keys and Certs?

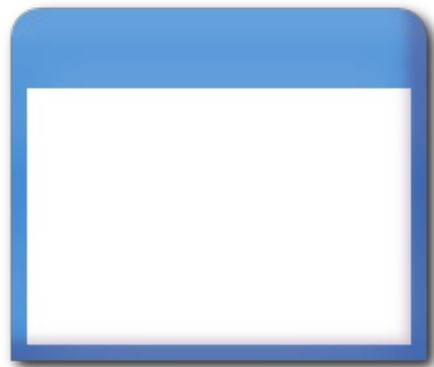


# PKCS#11 URIs

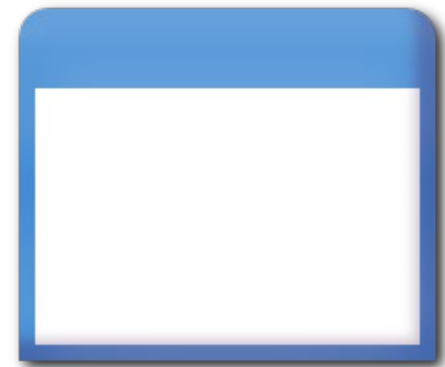
```
pkcs11:object-type=private;  
object=MyKey;  
token=Magic%20Token;  
id=%69%97%5c
```



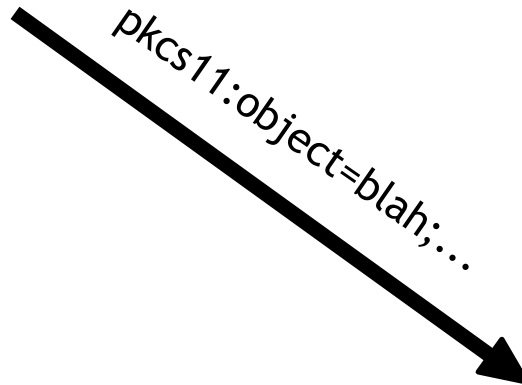
# PKCS#11 URIs



pkcs11:object=blah;...



pkcs11:object=blah;...



# RFC: PKCS#1 1 URI Scheme

<http://tools.ietf.org/html/draft-pechanec-pkcs11uri-03>



# Three steps....



Store keys and certificates interoperably



Make consistent trust decisions



Refer to keys and certificates in a standard way



# Any qvestions?

<http://p11-glue.freedesktop.org>

[p11-glue@lists.freedesktop.org](mailto:p11-glue@lists.freedesktop.org)

- Icon: <http://www.iconspedia.com/icon/smart-card-10843.html>
- Thanks to everyone who contributed!

