More secure with less "security"

Stef Walter

My name is Stef Walter. I work at Red Hat.
* I'm interested in making stuff just work.
Today I'm going to talk to you about making security just work.
First covering some abstract concepts
And then a few examples of implementing them
Interrupt if you want


“the user”

There's often talk of a mythical being
In the security community we chide the human for clicking on things, answering security questions incorrectly, choosing passwords that can be remembered, or the same password, or for wanting to install software. The user falls for phishing tactics etc...


hu·man
[hyoo-muhn or, often, yoo-]

Humans are intelligent, fun, creative, crazy
But humans are overwhelmed by choice in the world today
The user may be physically capable of learning about security...
But there's not a chance they're going to choose to
Book by Barry Schwartz

“Filtering out extraneous information is one of the basic functions of consciousness”

— Barry Schwarz

Talk about the paradox of choice
Don't be suprised when the user ignores something you wanted him to see

freedom != choice

Your human wants to be free
Your human wants to be empowered
The user thinks they want choice
But what the user wants is meta-choice
They want to be free to choose
The human doesn't want to micromanage, wants to make high level decisions as much as possible.

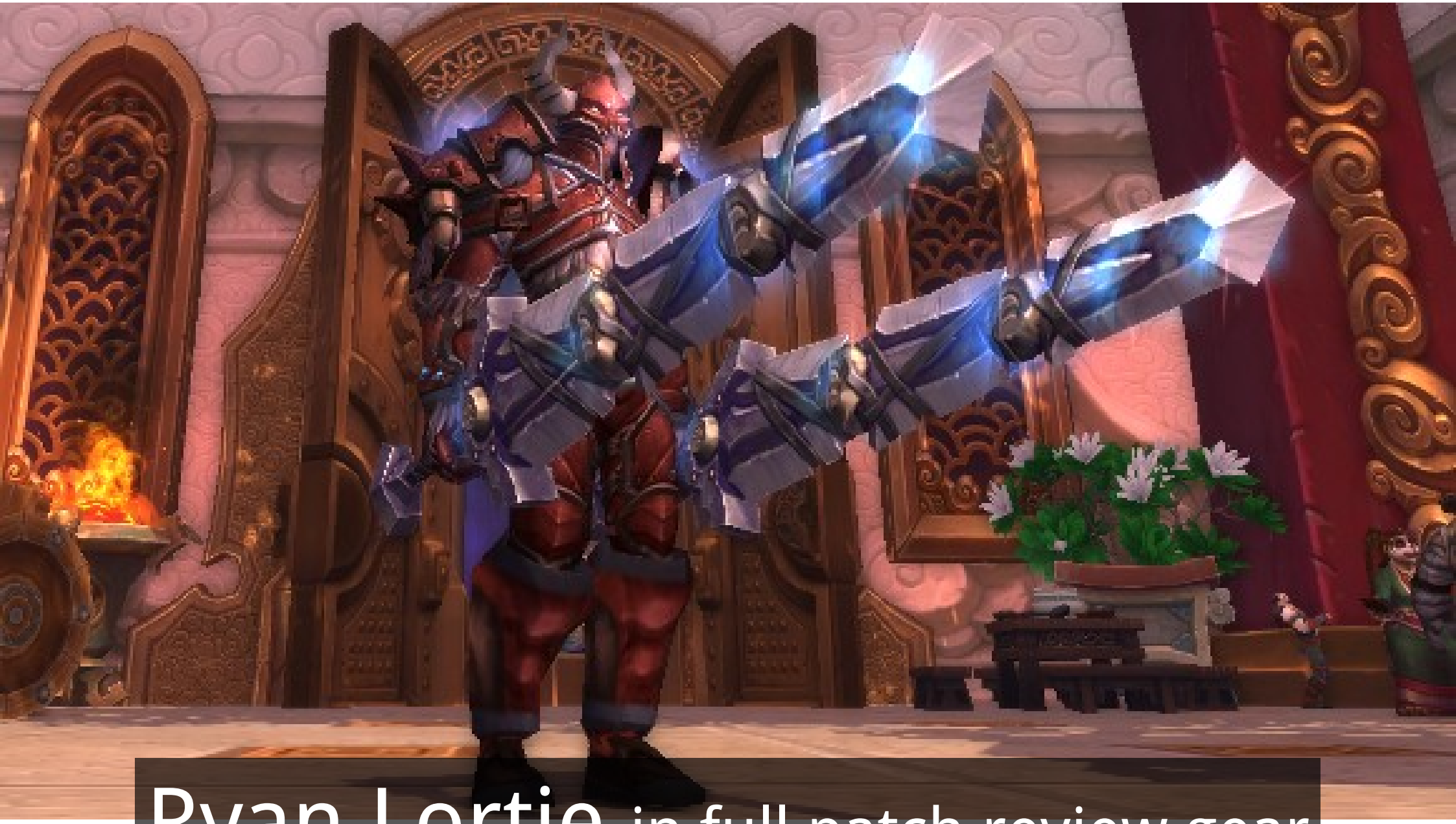

IF YOU FORCE THE USER TO BE A PART OF A SECURITY SYSTEM
INSTRUCTOR
YOU'RE GONNA HAVE A BAD TIME
quickmeme.com

People writing the software have a much better understanding
of the choices involved and how to
Talk about doctors and, the choices there

The extent of the human's involvement in security is to identify themselves

Professionals?

Professionals use different tools
A fireman doesn't try to turn his car into a fire engine



Ryan Lortie in full patch review gear

Professionals can be treated differently than humans
Embrace your inhumanity
May be professional in one area, but doesn't want to micromanage all areas.
A fireman won't use his truck to drive home or go on vacation

the worst possible time to ask a user a risky question?

when they're trying to do something.


worse than random chance.

If you flipped a coin you can get a better correct response in a risky situation.
But you can do better than either the user or random chance
You are aware of the trade offs. You're a professional.

Prompts are

dubious

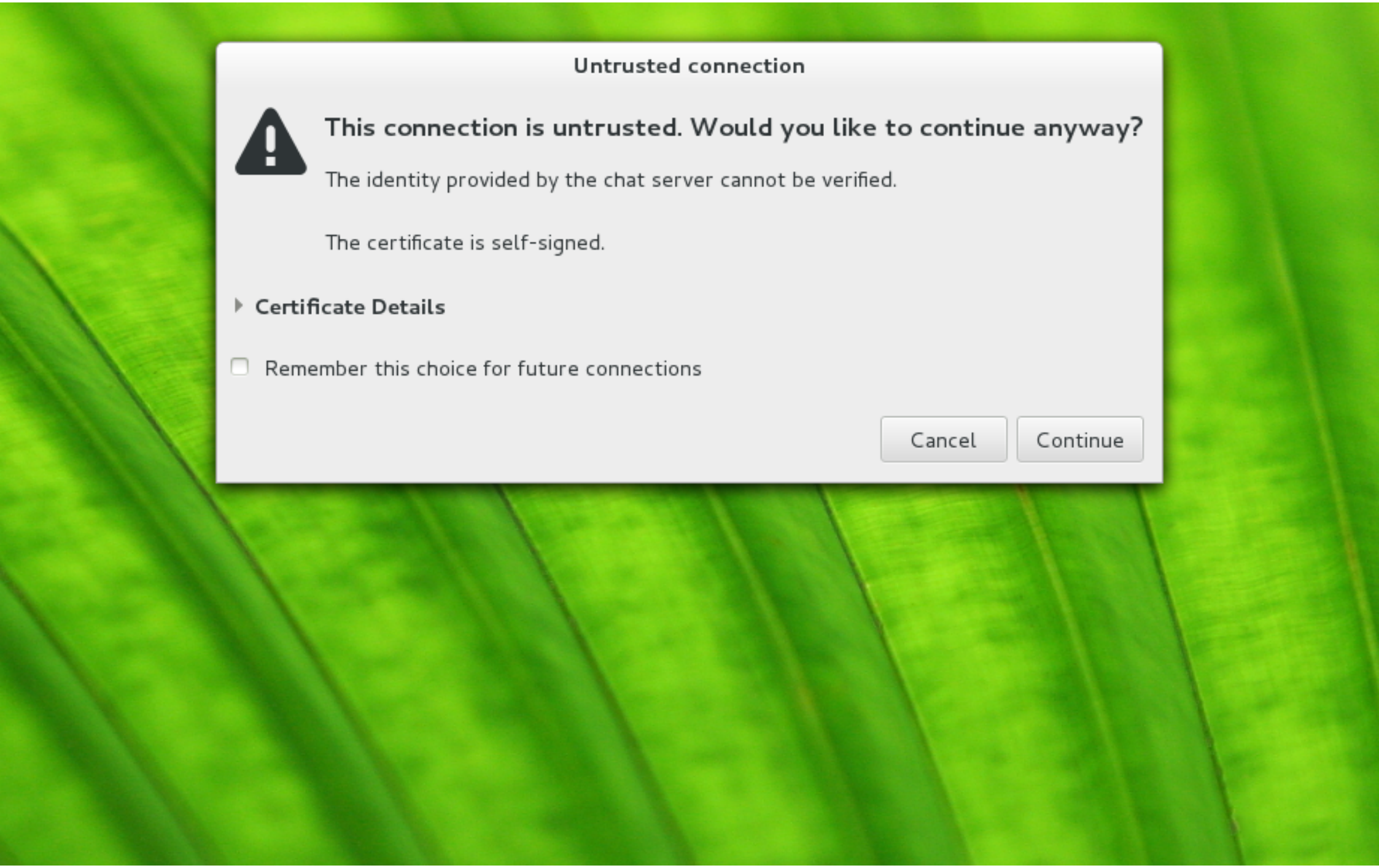
Security prompts are

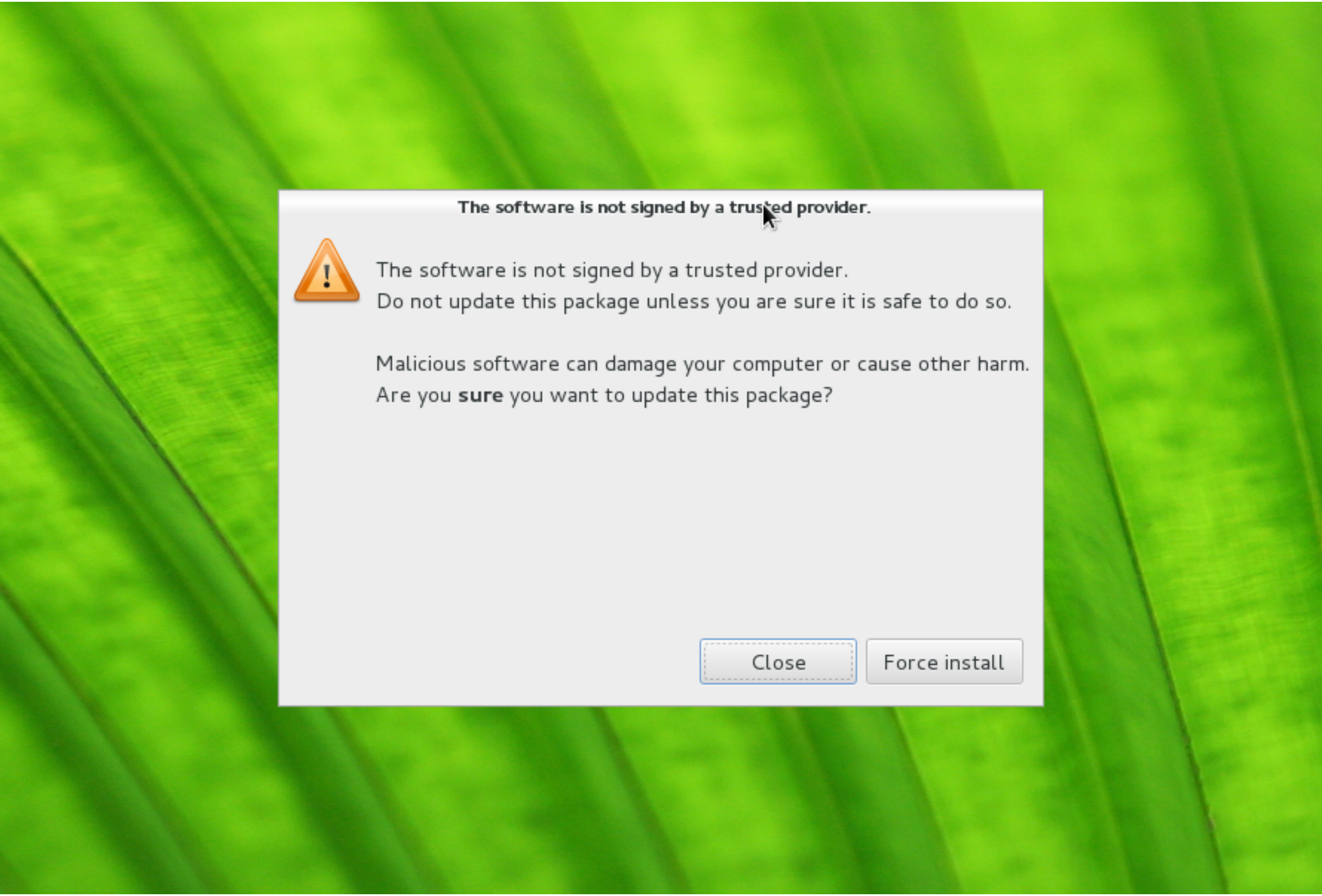
wrong

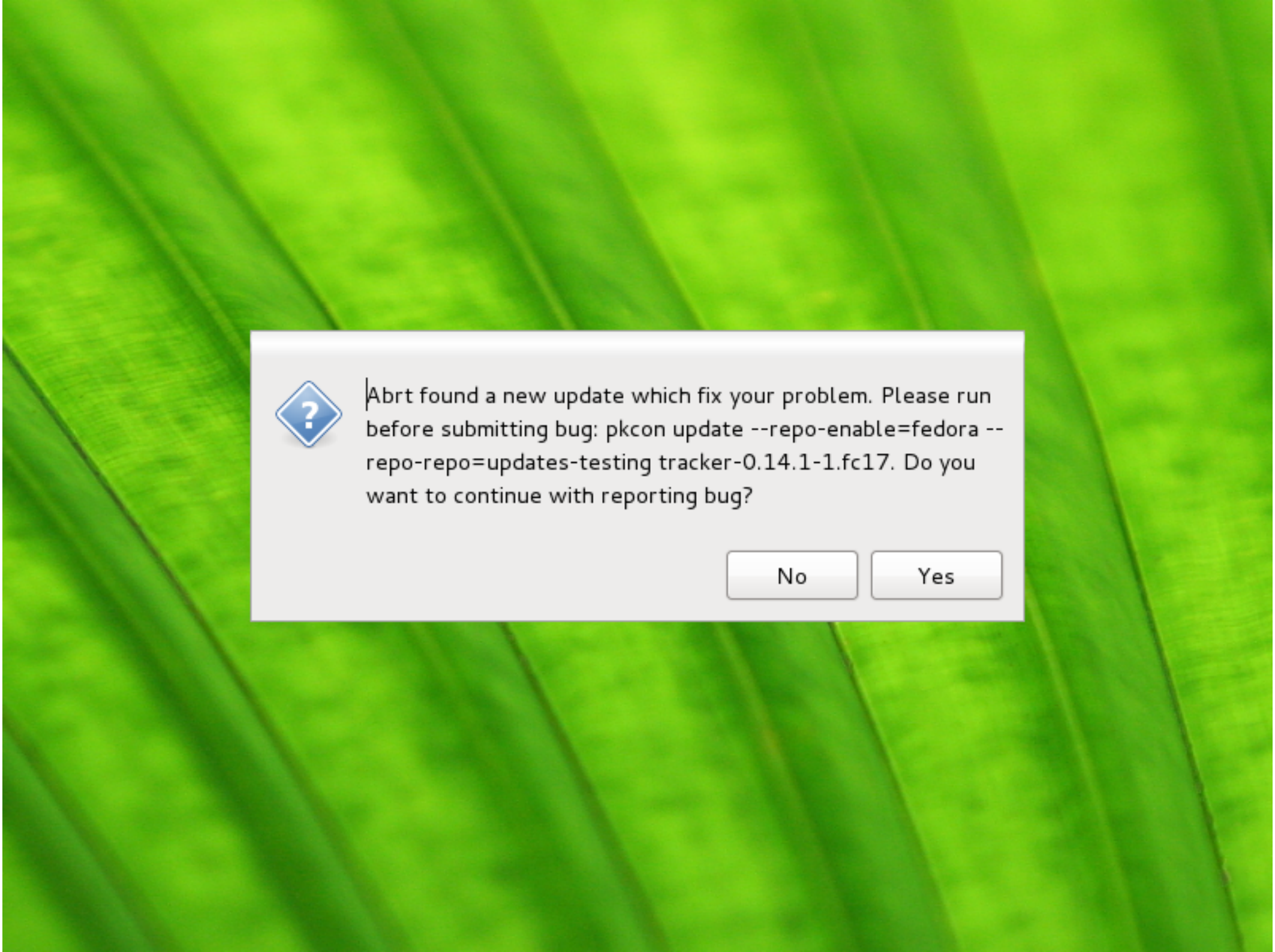
Sometimes you have to prompt for a user to identify themselves and we try to do that as little as possible.

Interrupting the user to make a permanent security decision is

EVIL


Untrusted connection
This connection is untrusted. Would you like to continue anyway?
The identity provided by the chat server cannot be verified.
The certificate is self-signed.
Certificate Details
Remember this choice for future connections
Cancel
Continue


The software is not signed by a trusted provider.
The software is not signed by a trusted provider.
Do not update this package unless you are sure it is safe to do so.
Malicious software can damage your computer or cause other harm.
Are you sure you want to update this package?
Close
Force install


Abrt found a new update which fix your problem. Please run before submitting bug: pkcon update --repo-enable=fedora --repo-repo=updates-testing tracker-0.14.1-1.fc17. Do you want to continue with reporting bug?
No
Yes


GAME
OVER
MAN

Game over, you lose

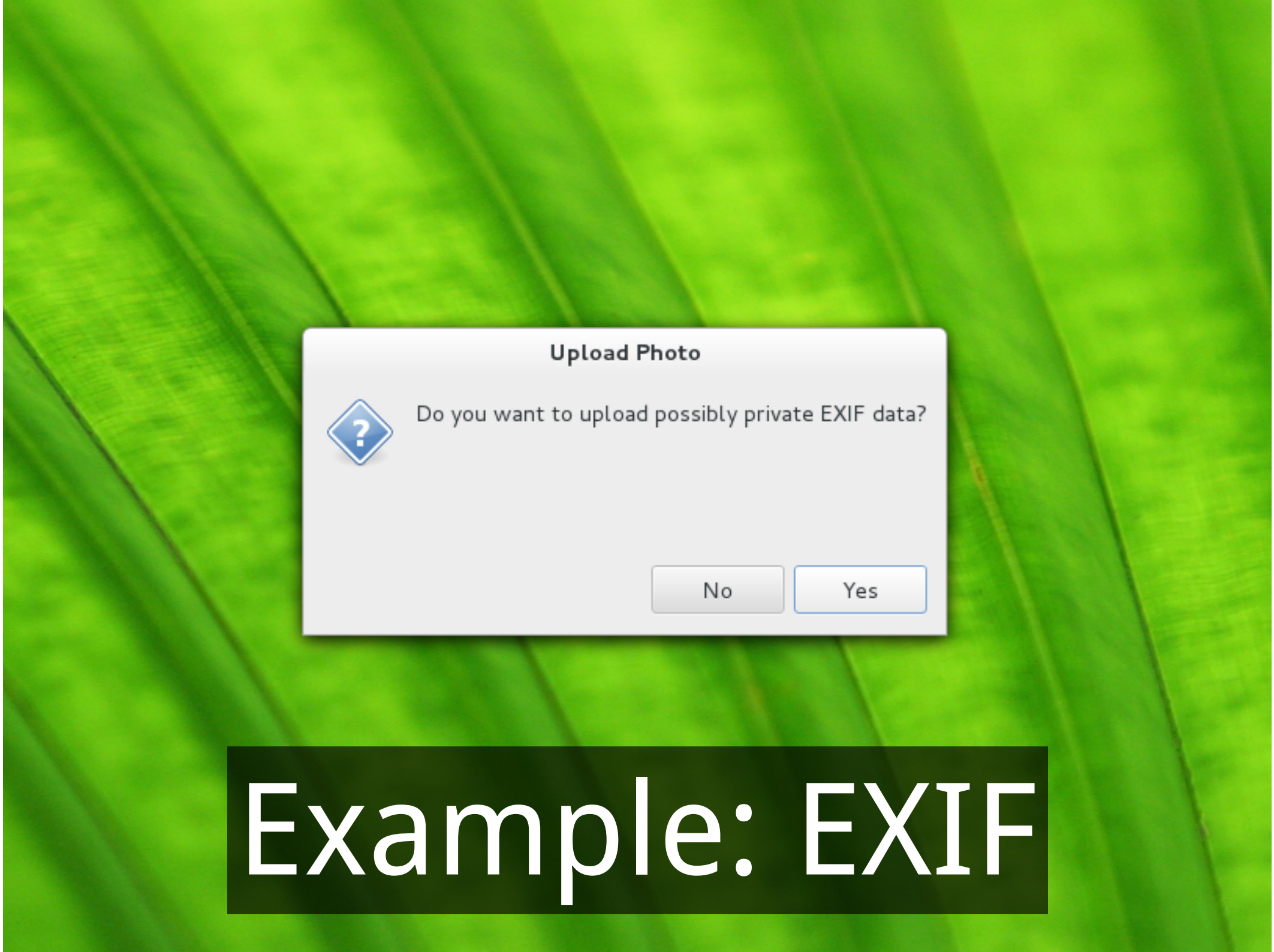
Stop interrupting

Let the user express their intent

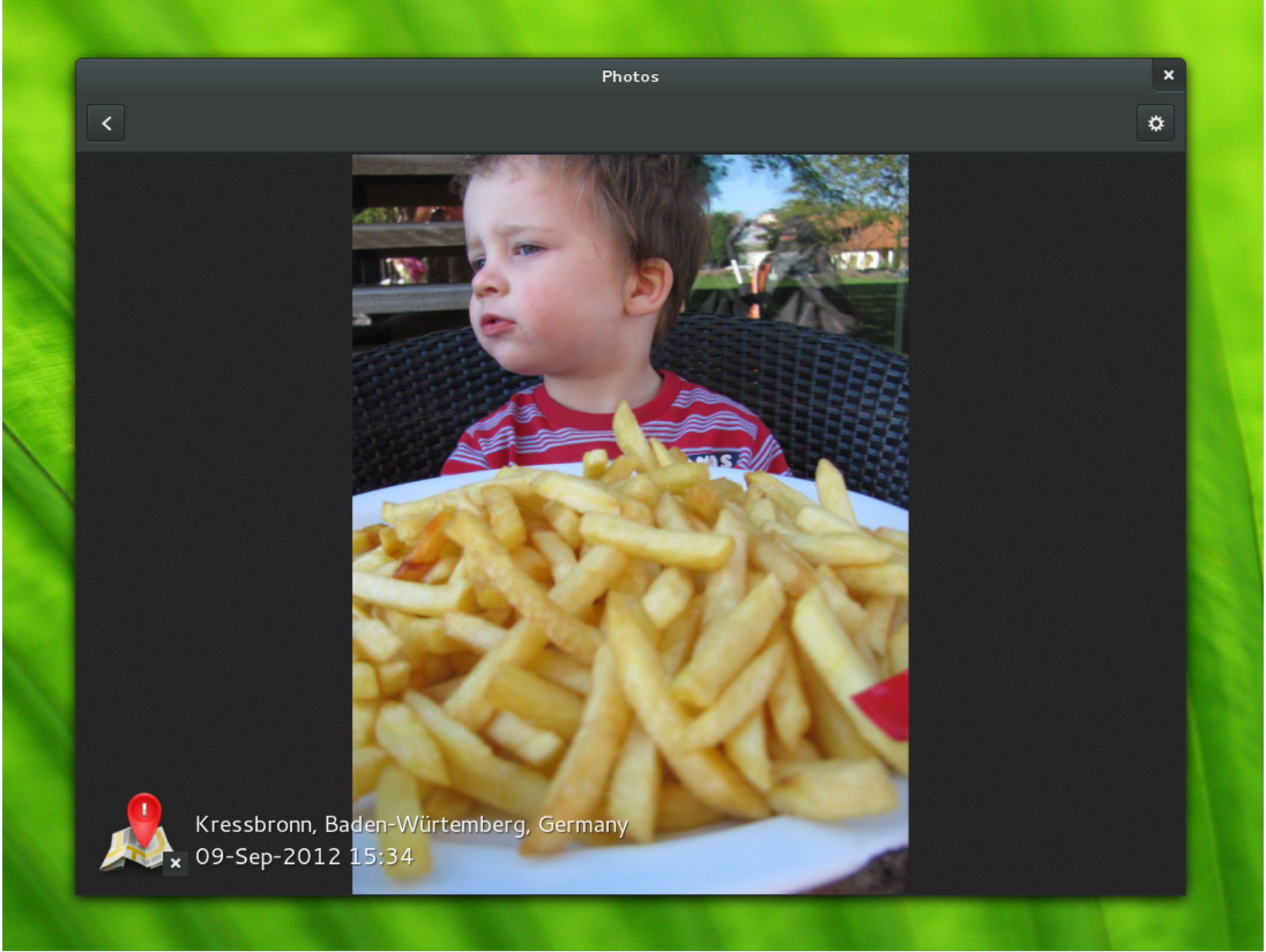
Let the user express their intent and take action based on that.

Example: Portals

Canonical example is the file chooser that's being discussed for sandboxked application.

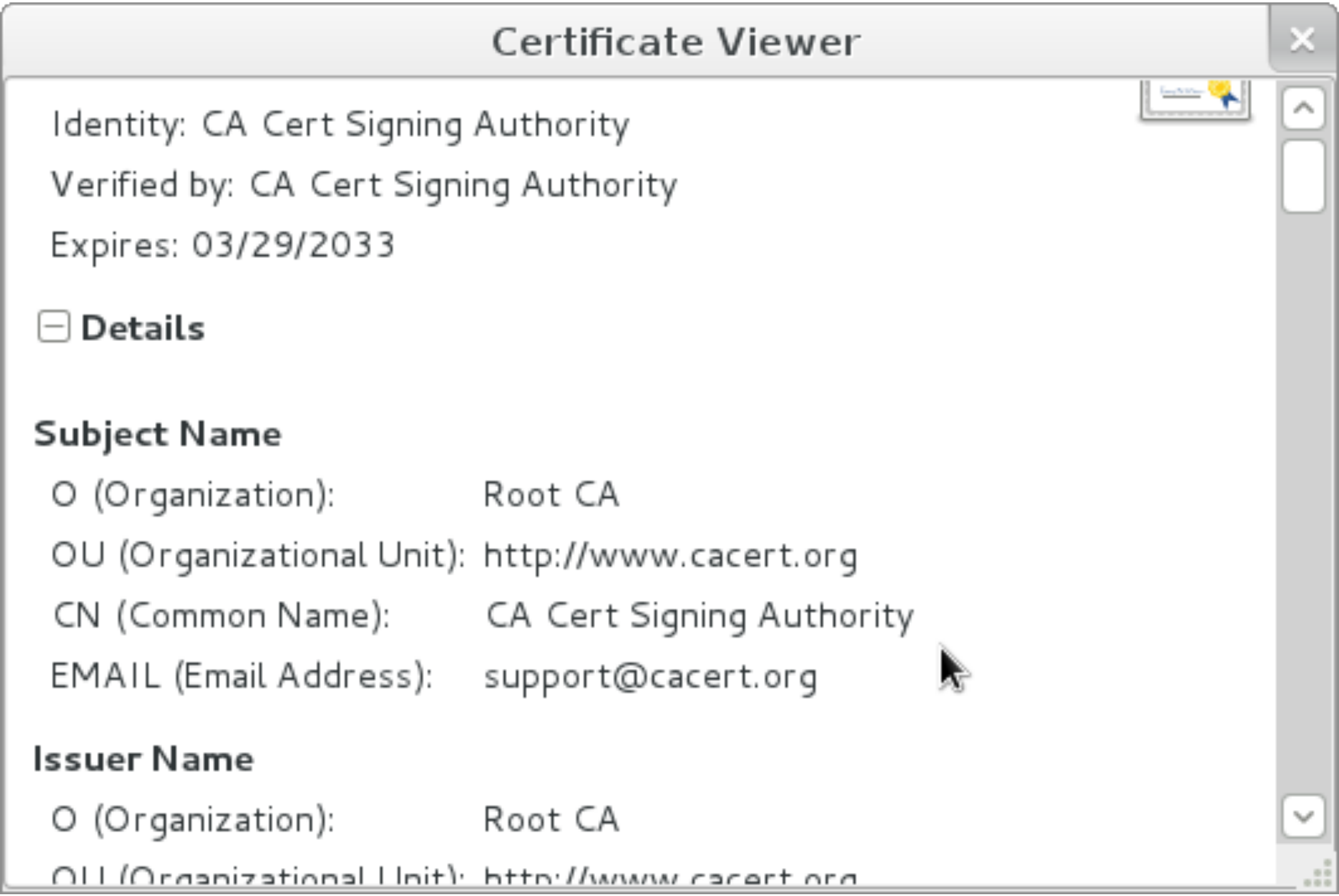

Upload Photo
Do you want to upload possibly private EXIF data?
No
Yes
Example: EXIF

Another example is a privacy feature of warning the user about uploading EXIF data


Photos
Kressbronn, Baden-Würtemberg, Germany
09-Sep-2012 15:34

Fix(ing) it!

Bye bye Certificate Prompts


Certificate Viewer
Identity: CA Cert Signing Authority
Verified by: CA Cert Signing Authority
Expires: 03/29/2033
Details
Subject Name
O (Organization): Root CA
OU (Organizational Unit): http://www.cacert.org
CN (Common Name): CA Cert Signing Authority
EMAIL (Email Address): support@cacert.org
Issuer Name
O (Organization): Root CA

The user is completely ill equipt to look at the details of a certificate and make a decision based on that.

Just drop the connection

But but but

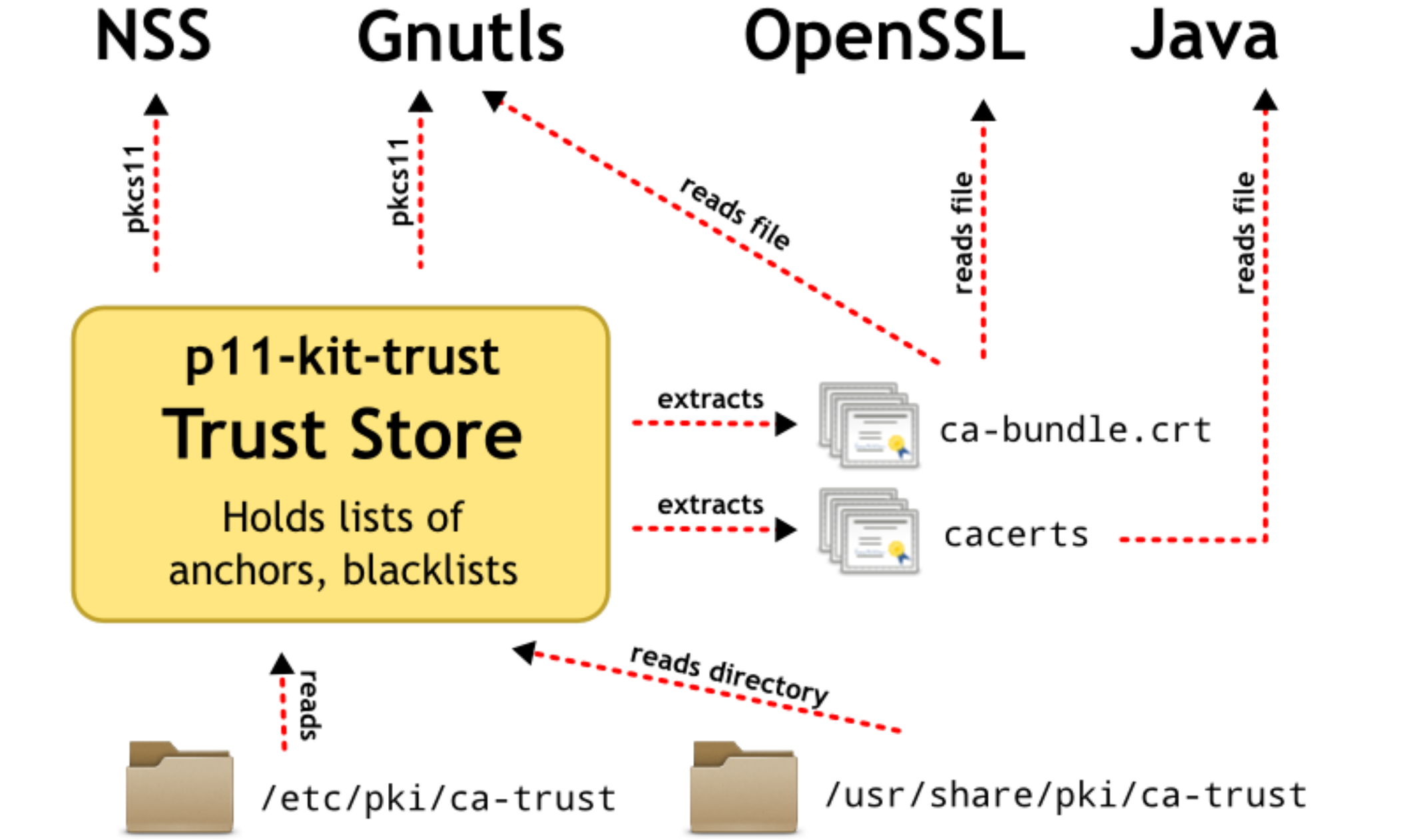
Enterprise users need to be able to use an Enterprise CA
We can do that.


Configure an Enterprise CA

Can now store anchors

We can now store anchors and blacklists globally so they're respected by all apps and

Shared System Certificates



PKCS#11 is the glue that makes this work.
Not all crypto libraries support PKCS#11 yet so we basically have to extract for some.

Professionals:

Pinning certificates to accounts

Make a pinned cert part of the account config, not global for the host.
This covers two use cases:
* Allowing use of development or misconfigured servers
* Micromanaging security, so you explicitly approve the certificate
  and want to be notified when it changes.
But your app doesn't have to do this if it's not a professional tool.

Application Password Storage

gnome-keyring is currently a central database of passwords
The user's intent is to share a password with an application
and is surprised when it shows up in a database readily available
after login by anyone who touches their computer.

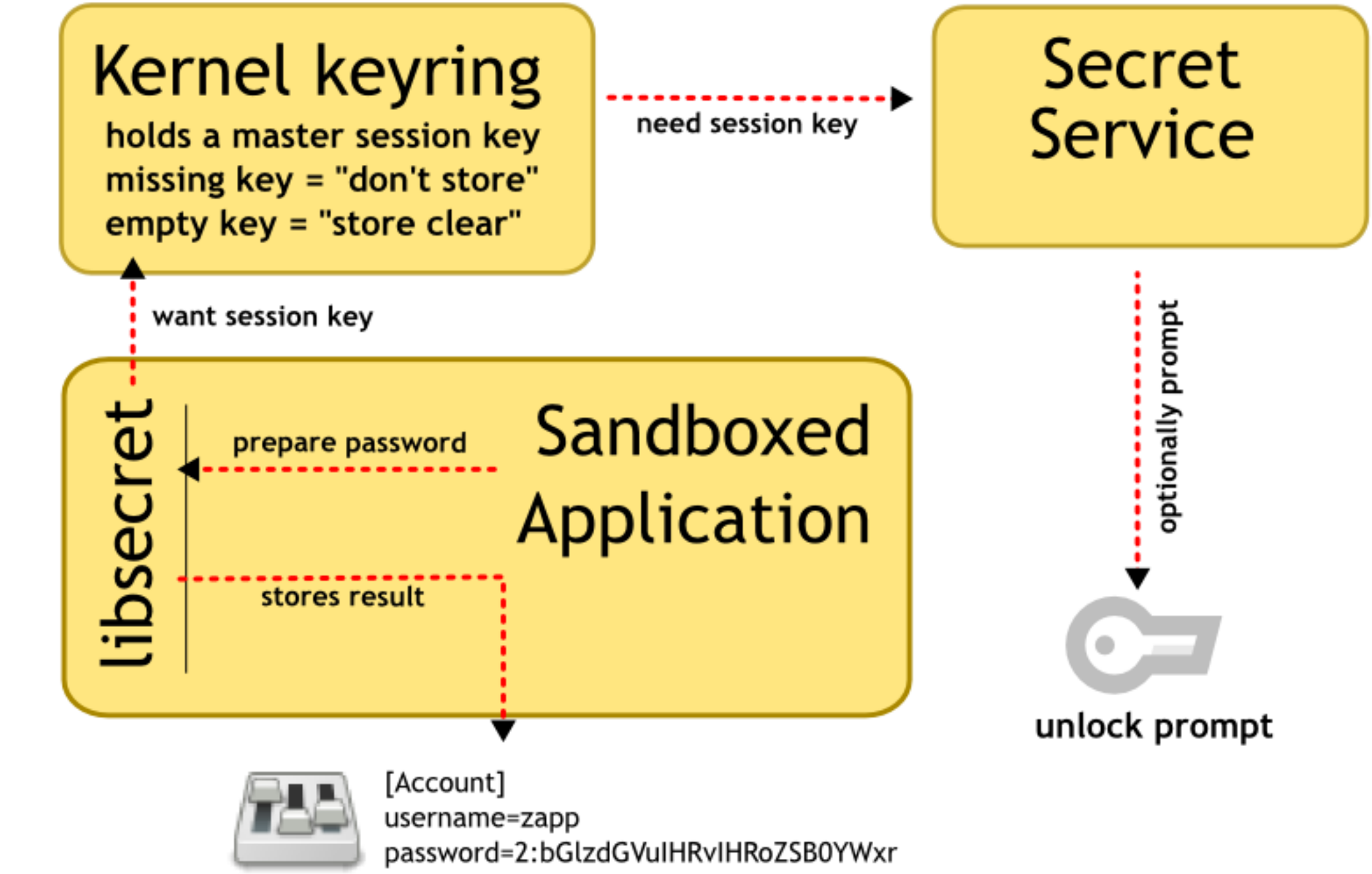
Password is part of account info

It should be stored as part of the account info.

Likely need encryption on disk

The reason we haven't stored it with the account info is because we want to keep the passwords encrypted on disk.
Some devices or machines have a secure disk (eg: encrypted) and in these cases storing passwords clear on disk is a-okay.

Application Password Storage

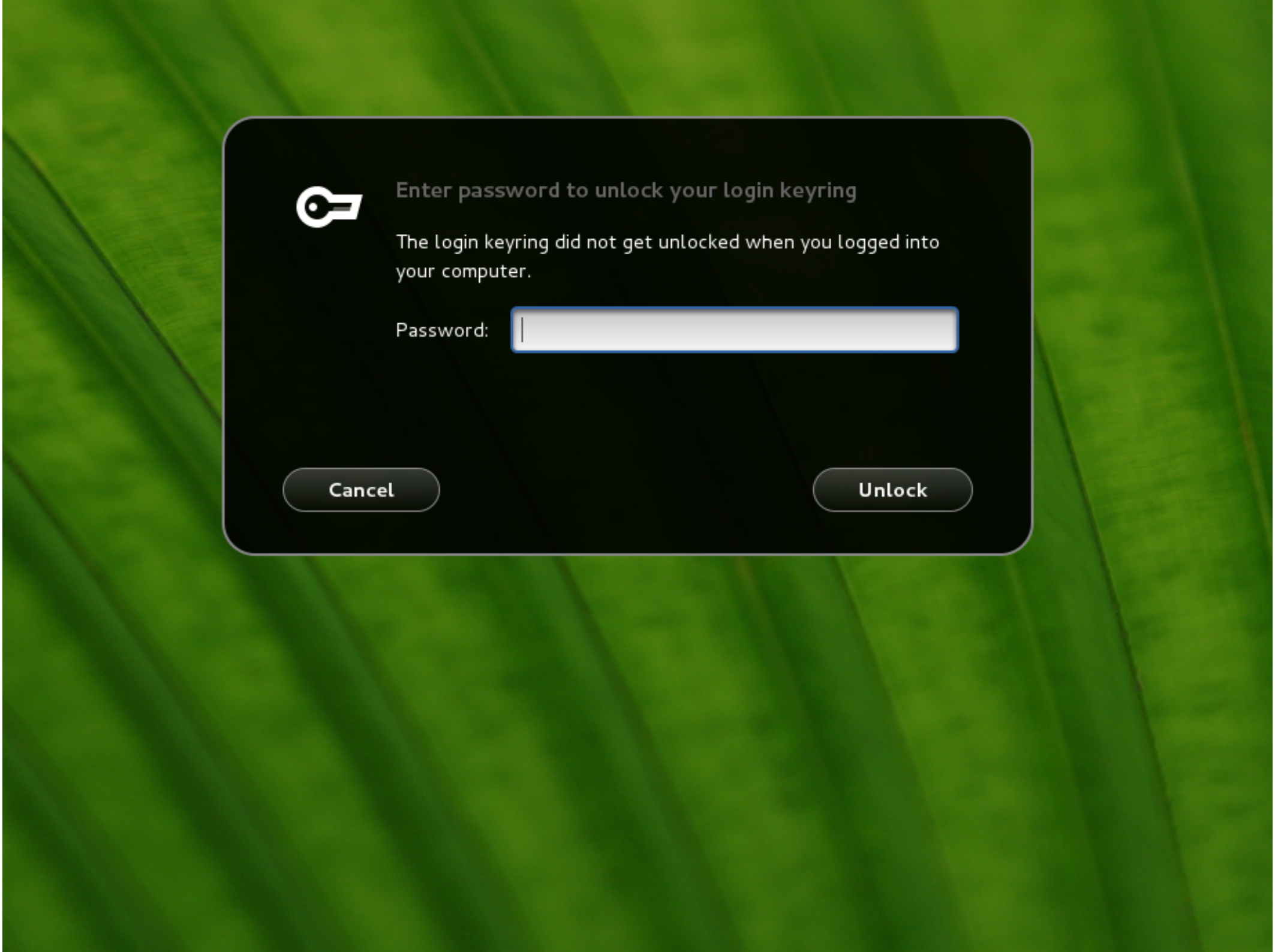


Talk about the kernel keyring
No surprises about where the passwords are stored
Matches user intent
Works with sandboxing
Null session key, means don't store the password
Empty session key means store the password in clear text

Perhaps: Archive passwords?

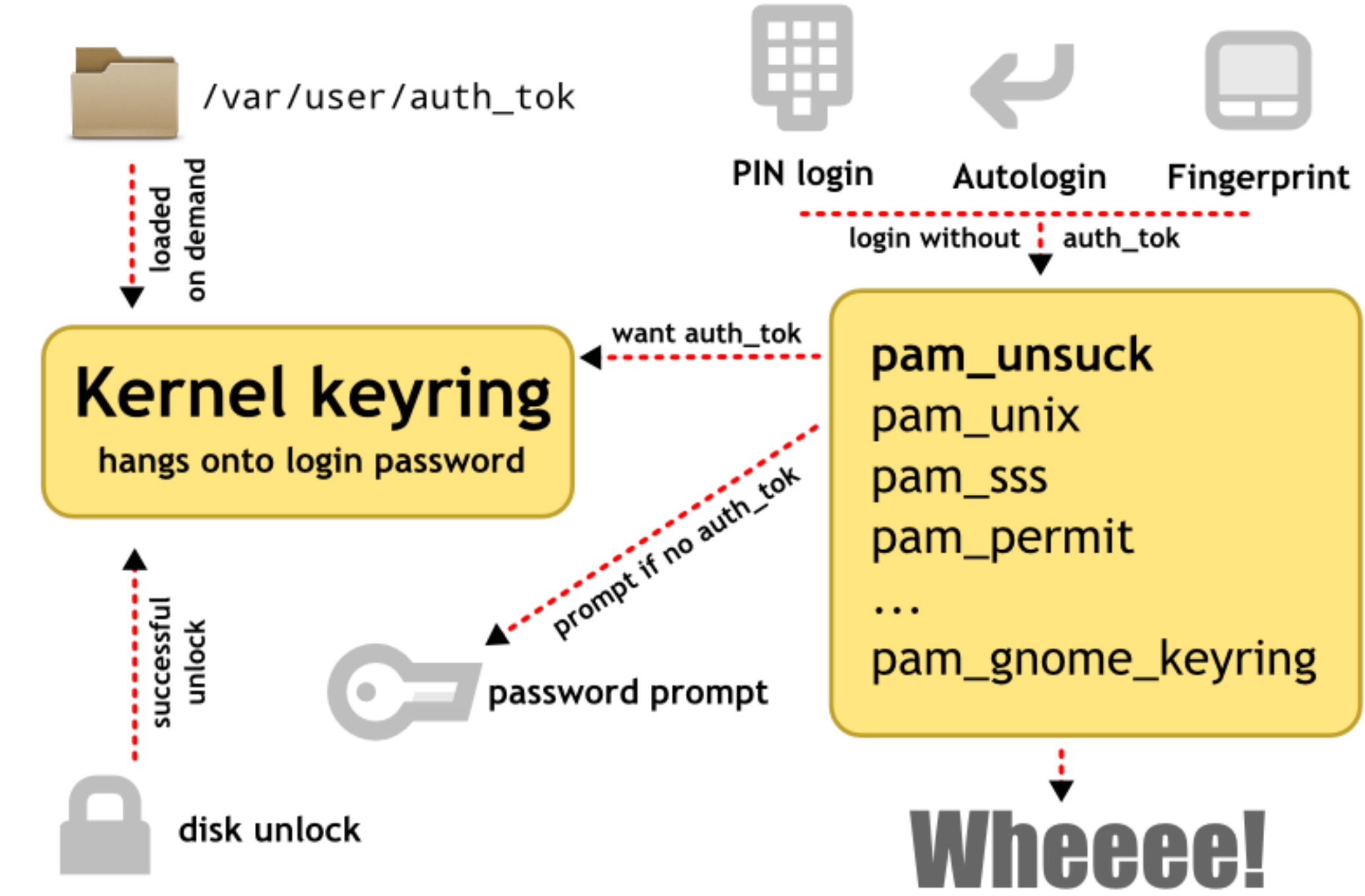
There is a secondary use case for a central database
and that is as a backup or lookup for passwords.
May still need to have something like this for password
archival, but design something specifically for this case

Unsuck
Login
Unlock


Enter password to unlock your login keyring
The login keyring did not get unlocked when you logged into your computer.
Password:
Cancel
Unlock

Problem, we see a password prompt at a password-less login
Make it possible to use fingerprints
Make it possible to use PIN when
Make it possible to auto login

Unsuck Login Unlock



The architecture described here
Would be nice the stored auth_tok to machine
using TPM chip or NVRAM


Join the dark side
You may be granted access to cookies

Go forth and kill

prompts

Any prompt should be regarded with suspicion
But terminate security yes/no prompts with extreme prejudice

Ellisons Law:

For every keystroke or click
required to use a crypto feature
the userbase declines by half.

Any Kvestions?

gnome-keyring-list@gnome.org
#keyring at gimpnet
http://p11-glue.freedesktop.org

stefw@gnome.org

Credits:

jimmac.musichall.cz
tychay at flickr.com
oliharwood at flickr.com
scradam at flickr.com
bitreaper1 at somethingawful.com
memegenerator.com